

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Комп'ютеризовані системи управ-  
ління»  
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»  
на тему: «Система моделювання поведінки натовпу»**

Виконала:

студентка IV курсу, групи ІА-61

Малецька Ольга Миколаївна \_\_\_\_\_

Керівник:

доцент кафедри АУТС, к.т.н.

Дорогий Ярослав Юрійович \_\_\_\_\_

Рецензент:

доцент кафедри ПЗКС, к.т.н.

Цуркан Василь Васильович \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2020

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматики та управління в технічних системах**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**  
**Малецькій Ользі Миколаївні**

1. Тема проєкту «Система моделювання поведінки натовпу», керівник проєкту Дорогий Ярослав Юрійович, к.т.н., доцент, затверджені наказом по університету від «7» травня 2020 р. №1081-с.
2. Термін подання студентом проєкту: 11 червня 2020 р.
3. Вихідні дані до проєкту: науково-технічна література про методики, що описує штучні нейронні мережі. Публікації про навчання агентів за допомогою нейромережевої системи. Матеріали з мережі Інтернет за темою створення поведінок агентів системи за допомогою штучних неронних мереж та їхнє подальше навчання за допомогою генетичного алгоритму.
4. Зміст пояснювальної записки: Зробити огляд існуючих рішень, обґрунтувати обрані технології та методи, розробити алгоритм та спроектувати систему.

5. Перелік графічного матеріалу: UML структура штучної нейронної мережі стратегії обходу перешкод, UML діаграма варіантів використання, UML діаграма основних класів програми, UML блок-схема роботи генетичного алгоритму.

6. Дата видачі завдання 05.03.2020 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованих матеріалів з теми	05.03.2020-15.03.2020	
2	Постановка задачі	05.03.2020-15.03.2020	
3	Огляд існуючих рішень	15.03.2020-20.03.2020	
4	Вибір методик та їх обґрунтування	27.03.2020-12.04.2020	
5	Розробка алгоритму та проектування	15.04.2020-30.04.2020	
6	Виконання графічних документів	02.05.2020-20.05.2020	
7	Оформлення документації	20.05.2020-01.06.2020	
8	Захист дипломного проєкту	16.06.2020	

Студент

Ольга МАЛЕЦЬКА

Керівник

Ярослав ДОРОГИЙ

## АНОТАЦІЯ

Малецька О.М. Система моделювання поведінки натовпу. КПІ ім. Ігоря Сікорського, Київ 2020.

Проект містить 65 сторінок тексту, 19 рисунків, 5 таблиць, посилання на 21 літературне джерело, 4 додатки.

Ключові слова: штучні нейронні мережі, система моделювання, навчання без учителя, агенти, поведінка.

Об'єктом розробки є система моделювання поведінки натовпу.

Мета розробки – покращення систем моделювання поведінки натовпу.

Даний дипломний проект присвячено моделюванню поведінки натовпу. Метою бакалаврської роботи є створення системи з унікальним функціоналом, за допомогою якого отримані результати можна використати в інструктажах з техніки безпеки при екстрених ситуаціях або при проектуванні громадських будівель.

В дипломному проекті розглянуті основні методи моделювання нейронних мереж для відображення поведінки натовпу, їх переваги та недоліки.

## ANNOTATION

O. Maletska. The system of modeling crowd behavior. Igor Sikorsky KPI, Kyiv, 2020.

The project contains 65 pages of text, 19 figures, 5 tables, links to 21 literary sources, 4 annexes.

Keywords: artificial neural networks, system of modeling, training without a teacher, agents, behavior.

The object of development is a crowd behavior modeling system.

The purpose of the development is to improve systems for modeling crowd behavior.

This graduation project is dedicated to modeling crowd behavior.

This graduation project is dedicated to modeling crowd behavior. The aim of the bachelor's work is to create a system with unique functionality, with which the results can be used in emergency training or in the design of public buildings.

The thesis project discusses the basic methods of modeling neural networks to display the behavior of the crowd, their advantages and disadvantages.

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Система моделювання поведінки  
натовпу»**

Київ - 2020

## ЗМІСТ

ВСТУП .....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	6
1.1 Теоретичні відомості про поведінку натовпів .....	6
1.2 Поняття агентного моделювання .....	9
1.3 Область застосування імітації дійсності.....	11
1.4 Проблеми агентного моделювання поведінки.....	13
1.5 Засоби реалізації систем агентного моделювання.....	14
1.6 Опис існуючих платформ агентного моделювання.....	14
1.7 Існуючі рішення агентного моделювання натовпу в екстремальних ситуаціях. ....	16
2 АНАЛІЗ ІСНУЮЧИХ СТРУКТУР НЕЙРОННИХ МЕРЕЖ ТА МЕТОДІВ НАВЧАННЯ .....	19
2.1 Штучні нейронні мережі та способи їх побудови .....	19
2.2 Огляд штучного нейрона.....	20
2.3 Архітектура штучних нейронних мереж .....	23
2.3.1 Одношаровий персептрон .....	25
2.3.2 Багатошаровий персептрон.....	26
2.3.3 Рекуррентні мережі.....	27
2.4 Навчання нейронних мереж.....	29
2.4.1 Навчання з учителем та без.....	30
2.4.2 Генетичний алгоритм навчання.....	31
2.5 Застосування генетичного алгоритму та штучних нейронних мереж в розробках з моделювання поведінки об'єктів .....	34

					ІА61.140БАК.005 ПЗ			
Зм.	Лист	№ докум.	Підпис	Дата	Система моделювання поведінки натовпу  Пояснювальна записка	Літ.	Арк.	Акрушів
Розроб.		Малецька О.М.						
Перевір.		Дорогий Я.Ю.					2	65
						«КПІ ім. І.Сікорського» ФІОТ, група ІА-61		
Затверд.								

3 РОЗРОБКА МЕТОДИКИ ТА СИСТЕМИ МОДЕЛЮВАННЯ ПОВЕДІНКИ НАТОВПІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ ТА ГЕНЕТИЧНОГО АЛГОРИТМУ .....	36
3.1 Розробка системи, що керує модельованими об'єктами .....	36
3.2 Огляд стратегії поведінки модельованого об'єкта.....	37
3.2.1 Розробка початково необхідного переліку можливих типів поведінки до стратегій та тактик .....	38
3.2.2 Розробка критеріїв підбору поведінок за допомогою генетичного алгоритму .....	41
3.3 Розробка типів поведінки на основі штучної нейронної мережі .....	45
3.3.1 Штучна нейронна мережа, що дозволяє об'єкту дістатися до кінцевої точки, оминаючи вогонь та перепони .....	45
3.3.2 Навчання мережі генетичним алгоритмом.....	49
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОЇ СИСТЕМИ.....	52
4.1 Оцінка ефективності запропонованої системи та її програмної реалізації.....	52
4.1.1 Опис реалізації системи вцілому .....	52
4.1.2 Дослідження роботи системи моделювання натовпів .....	54
4.1.3 Реалізація алгоритму навчання модулю на нейронній мережі .....	57
4.2 Дослідження роботи алгоритму.....	57
4.3 Пропозиції щодо програмної реалізації системи моделювання поведінки натовпу під час пожежі .....	61
ВИСНОВКИ.....	63
ПЕРЕЛІК ПОСИЛАНЬ ТА ВИКОРИСТАНИХ ДЖЕРЕЛ .....	65



## ВСТУП

Нейронні мережі та різні алгоритми їхнього навчання широко використовуються у розпізнаванні та відтворенні образів, тенденцій, послідовностей, класифікації даних, в моделюванні систем, що імітують агентську поведінку.

Використання цього підходу дозволило б створити моделі людей, які перебувають у натовпі в натовпі та мають поведінку близьку до реалістичної. Це дозволить ширше дослідити натовп та проводити експерименти з навченими моделями. Такі дослідження потрібні при проектуванні будівель, що вміщують велику кількість людей, як концертні зали, виставкові центри, метро, аеропорти. Вони проводяться зокрема для правильного встановлення екстрених виходів у разі виникнення паніки з забезпечення ефективної евакуації.

Метою даної роботи є дослідження та створення методики та тестового застосування, що моделює евакуацію натовпу з приміщення під час пожеж: У 3-вимірній моделі великого приміщення з проходами, завадами, стінами та виходами знаходитимуться об'єкти у вигляді кубів, що рухаються процедурно навмання.

У певний момент деякі частини кімнати будуть займатися полум'ям. Перед кубічними об'єктами стоятиме завдання покинути приміщення при цьому уникаючи вогню та не створюючи затори на виходах із приміщення, вони мають це зробити базуючись на даних про попередні події з минулих ітерацій, так як поведінка та результат дій об'єктів записуватиметься до бази даних.

Тобто очікується, що спочатку, при перших запусках цього процесу об'єкти нездатні будуть покинути кімнату так як у них буде недостатньо даних про те як не уникати вогню, не створювати натовпи та інше, проте на

пізніх ітераціях вони зможуть організовуватися в «чергу» та не перешкоджати один одному з метою збереження більшої кількості об'єктів від вогню.

Метою є навчання об'єктів на їхньому власному досвіді будувати свою поведінку виходу з занятого приміщення, властивості та наповнення якого змінюється від ітерації до ітерації за допомогою нейронних мереж.

Штучні нейронні мережі використовуються в якості регулятора багатовимірних і багатозв'язних динамічних об'єктів. Застосування штучних нейронних мереж для цілей управління є однією з численних областей відносного нового розділу сучасної науки — нейроінформатіки, змістом якої є розробка і дослідження методів вирішення найрізноманітніших завдань за допомогою штучних нейронних мереж, побудованих на стандартних штучних нейронах.

Як один з найбільш ефективних підходів моделювання, буде запропоновано створення поведінок агентів системи за допомогою штучних неронних мереж та їхнє подальше навчання за допомогою генетичного алгоритму. Багатоагентна система представляє собою систему, що утворюється із певної кількості самостійних елементів, котрі взаємодіють між собою та реагують один на одного. Агент буде представляє собою певну програму або алгоритм дій. Така послідовність містить достатні та повні запрограмовані правила агентної поведінки.

Такий підхід моделювання набуває ефективності у тому випадку, якщо відсутні інші варіанти спостереження за агентами та їхніми діями. Часто це відбувається по причині того, що такий процес є довготривалим (наприклад, протягом створення нової мови, що займає кілька сотень років в реальному часі або небезпечним для реальних агентів. Також після розгляду інсуючих рішень з моделювання поведінки живих організмів та обрання підходящої структури нейронної мережі буде розроблено такий модуль поведіки, котрий буде включено до інших модулів та зроблено висновки про його ефективність у порівнянні з іншими.

Система та нейронна мережа мають працювати в двох режимах: режим навчання і режим функціонування. Для режиму навчання використовуються дані експертної настройки регулятора. У режимі функціонування нейронна мережа повинна забезпечувати необхідну якість системи, шляхом зміни параметрів регулятора, в залежності від параметрів об'єкта.

Для вирішення проблеми, описаної у даній роботі використання штучних нейронних мереж досить бажане так як необхідно моделювати людську поведінку та прийняття рішень схожих до людських, а отже те що система рішень та обчислень схожа до людського мозку, а інструмент навчання бере свій початок у еволюційній теорії є значним плюсом.

Виконуватися застосування має при використанні 3D рушія Unity3D, котрий дозволяє швидко та зручно змоделювати форми та рух тривимірних об'єктів, що сприяє вигляду моделювання ситуацій. Буде побудовано модель СУБД, графічний інтерфейс до застосування.

Отже, корисними методика та застосунок є тим, що навчені моделі можуть використовуватися для перевірки можливості евакуації в реальних будівлях.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 Теоретичні відомості про поведінку натовпів

Натовп – це тимчасове об’єднання великої кількості людей, які мають безпосередній контакт між собою та майже ідентично реагують на певні стимули [1].

Вчені виділяють наступні етапи трансформації великого скупчення людей в емоційну спільність, особливу «масу».

Перший етап — циркуляція. Під час цього етапу у більшості людей, що знаходяться у натовпі з’являються загальні емоції, що передаються між учасниками групи та рухаються, циркулярно, повертаючись до попередніх вже підсиленими. На протязі часу така загальна емоція циркулює, проте якщо процес або подія що стала її джерелом, переривається, то вона з плином часу зникає.

Другий етап — кружіння. Даний етап виражає «емоційне кружіння» певного психофізичного стану, а не тільки емоції, як відбувалося в минулому стані. У стихійно створеній соціальній групі емоція рухається серед людей ніби по колу, безперервно підтримуючи та посилюючи саму себе. Почуття цієї емоції загострюються, зростає психоемоційне збудження. Поведінка та емоційний стан індивідів вже залежить у більшій мірі від емоцій оточуючих, а не формується раціонально інтерпретацією обставин, що створилися навколо. Підтримка та розвиток емоцій посилюються завдяки появі нових індивідів, які мимоволі приймають цей стан та починають приймати участь у його розповсюдженні. Посилення емоційного кружіння викликає різке зниження критичності суджень та підвищення навіюваності у учасників соціальної групи. Навіюваність зростає по відношенню до впливів, що йдуть зсередини спільноти, при цьому учасники вже не здатні сприймати раціональні повідомлення, що отримуються ззовні. Спільнота стає ізольованою

та самотійною в емоційному плані. Група людей може повністю переродитися в однорідну аморфну масу, яка несвідомо та згуртовано відповідає на зовнішні стимули. На цьому етапі учасникам притаманна готовність до дій та вчинків, що приймаються спільно.

Третій етап — створення образу. Емоційні імпульси, почуття та уява групи фокусуються на новому об'єкті загальної уваги. Якщо спочатку групу цікавила хвилююча подія, то на даному етапі новим об'єктом стає образ, створений у попередньому етапі та частково спілкування учасників групи. Теперішній образ приймається усіма членами спільноти, дає їм загальну орієнтацію на певні дії та згуртовує спільноту разом, утворюючи ціле. У процесі «емоційного кружіння» та «циркулярної реакції» зростає емоційна напруга. Учасники не здатні сприймати раціональні повідомлення, що отримуються зовні. Відбувається різкий розподіл на тих, хто з нами — «ми», та тих, хто проти нас — «вони». Виникає потреба у спільних діях для підтримання згуртованості.

Четвертий етап — активізація. Останній етап формування суб'єкта стихійної поведінки такої як натовпу — члени групи починають активізуватися. Цей процес проходить під додатковим емоційним стимулюванням та можливо по причині впливу лідера, якщо такого було знайдено. Лідер безпосередньо підштовхує учасників соціальної групи до конкретних, потрібних йому дій. Таке навіювання обов'язково має відповідати груповому, спільному образу, що сформувався у маси людей раніше. Стан напруження та збудження, в якому знаходиться натовп, знаходить свій вихід у негайних діях. Якщо лідера немає, група стихійно починає безпосередні дії щодо обраного об'єкта. Наблюдения и описания многочисленных эпизодов коллективной паники позволили вычленить некий «усреднённый» сценарий.

Шокуючий стимул, дуже сильний або той що повторюється, лякає спочатку одного або кількох учасників соціальної групи. У натовпі мінімальний поріг збудженості мають жінки або діти а в ситуації войовничій або бойовій

— молоді і не загартовані в боях солдати. Їхній стрес та переляк утворює односкладові крики (напр. «Пожежа!»), вираз облич і метушливі рухи тіла.

Ці учасники перетворюються на джерело страху від яких він передається іншим. Починається взаємна індукція і підсилення емоційної напруги завдяки механізмам циркулярної реакції. В разі не прийняття своєчасних заходів, натовп починає деградувати, люди втрачають самоконтроль, і виконують панічні дії, найчастіше прибігають до втечі, яке здається рятівним, хоча в дійсності тільки підвищує небезпеку.

Наявність в панікуючій масі жінок і дітей, про пріоритетний порятунок котрих в масовій паніці вже ніхто не думає, погано впливає на масу по причині того, що голос високої частоти, як і інші такі звуки, ще більше наводять паніку серед людей.

Гарний приклад спонтанних перетворень можна привести на прикладі футбольного матчу. З початком гри конвенціональний натовп стає експресивним і найважливіше завдання організаторів події — вжити всіх необхідних заходів, для запобігання перетворення натовпу на агресивний, та не допускати ситуації, коли тисячі уболівальників одночасно спрямовуються до єдиного виходу або не допустити екстремалу ситуацію, а саме спровоковану бійку, пожежу, стрілянину та інше.

Розглянемо ситуацію, яка сталася в «Лужниках», популярна московська команда зустрічалася з закордонним клубом. Результат був дуже важливим, але гра протікала досить мляво. Глядачі втрачали інтерес і, головне, надію на те, що вирішальний гол їх улюбленцями буде забитий. Вже за 15-20 хвилин до кінця люди почали залишати трибуни, деякі встигли вийти за ворота, інші перебували на шляху до виходу. І раптом, на останній хвилині матчу, довгоочікуваний гол був таки забитий, викликавши бурхливу реакцію уболівальників. Ті хто встиг покинути стадіон, кинулися назад, щоб взяти участь в загальній радості, і зіткнулися у вузькому проході з тими, хто, під

тиск задніх рядів, по інерції продовжував рухатися до виходу. Це обернулося трагедією: десятки людей були задавлені на смерть.

В історії є випадок, коли в 1938 році на трибунах паризького національного велодрому після закінчення змагання виникла невелика пожежа. Співробітникам вдалося швидко локалізувати вогонь, але десять тисяч глядачів з надмірною енергійністю вже рушили до єдиного виходу. Ситуація загрожувала стати смертельно небезпечною. Завдяки щасливому випадку в натовпі виявилися двоє психологів, які змогли вчасно зорієнтуватися і почали голосно скандувати: «Ne pousse pas!» (Не пус па — Чи не штов хай). Ритм підхопили навколишні, він пройшов по натовпу. Через кілька хвилин тисячі людей дружно скандували цю фразу; натовп перетворився на експресивний, страх і суєта змінилися загальним запалом і всі спокійно залишили трибуни [2].

## 1.2 Поняття агентного моделювання

Системи, що імітують поведінку живих організмів та процесів базуються на і, котре в свою чергу бере початок від агентного моделювання, і створюється із застосуванням штучного інтелектуального агента. Дане агентне соціальне моделювання представляє собою науку, пов'язану із моделюванням соціальних явищ, що використовує кібернетичні багатоагентні моделі для передбачення, моделювання та інших досліджень процесів. У таких моделях, агентами представляються одиниці моделювання або групи таких одиниць. Суміш соціальної наукової дисципліни, цифрового моделювання та багатоагентної моделі являє собою багатоагентне соціальне моделювання.

Агентні системи моделювання моделюють різні елементи соціальних систем. Задля спостереження за поведінкою агентів, застосовують застосовують інтелектуальних агентів, яких ставлять у комп'ютерну симуляцію соціуму.

					IA61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		9

За допомогою отриманих даних можна виокремити та дослідити різні реакції та відовіді інтелектуальних агентів. Як наслідок, трансформувати змодельовані реакції у результати реальних оригіналів моделювання та також реальних соціальних та природніх явищ та процесів.

Дизайн моделей та агентів називають агентним обчисленням. Моделювання агентів та наслідків, частково, представляє собою комп'ютерне моделювання. Суспільні науки вивчають та виявляють сполучення соціальної та наукової частин моделі. У цьому випадку, розвитку суспільних явищ, для їх моделювання створюють теорії. Для моделювання соціальних явищ, необхідно забезпечити програму моделюми та інструментами. У цьому полягає основна ціль агентних систем. З допомоги агентних систем моделювання з'явилася можливість дослідження наслідків явищ, котрі, можливо не має моливості спостерігати в реальному житті. Це надає дослідникам важливу інформацію щодо суспільства або наслідків соціальних подій або процесів.

Певна кількість, взаємодіючих між собою агентів, що реагують один на одного, являє собою багатоагентну систему. Такі елементи є агентами. Під час моделювання агентів можна використовувати для побудови багатьох різних елементів. Таких як, соціуми або машини, організми, особи або будь-які інші активні елементи. Вони можуть бути, реальними або навіть не існуючими. Агент, у багатоагентній системі, є певною програмою або алгоритмом дій. У такій послідовності містяться запрограмовані правила агентної поведінки, у достатньому та повному обсязі.

Агентні моделі, у спробах відтворити та передбачити виникнення складних феноменів, використовувть імітацію одночасних операцій та взаємодій множини агентів. Таким чином, зрозуміло, що це різновид мікро та макромасштабних моделей. До цього процесу можна застосувати принцип емерженості, при чому від найнижчого (мікро) рівня систем до найвищого (макро) рівня. Таким чином, виділяється ключове поняття: складна поведінка генерується простими правилами, цієї поведінки.

					IA61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		10



Існує принцип проектування, що відомий, як K.I.S.S. ("Keep it simple, stupid"). Він говорить, про те, що для найкращої роботи систем необхідно, щоб вони залишалися простими, а не ускладнювалися. Це тісно пов'язано з моделювальним середовищем. Ще одне з головних понять: загальне являється більшим, ніж сума його елементів. Це головне твердження. Індивідуальні агенти мають репродуктивну здатність, економічну вигоду чи соціальний статус. Це можна узагальнити, як здатність діяти у власних інтересах та характеризувати агентів як обмежено раціональних. Агенти систем моделювання можуть бути навчені адаптації, мати різні функції, такі як репродукція та різні види взаємодії з іншими агентами.

### 1.3 Область застосування імітації дійсності

Агентне моделювання в нашій звісно поширене в сфері біології, а також для аналізу поширення епідемій, аналізу можливостей виникнення різноманітної біологічної зброї, прогнозуванню виникнення біологічних проблем, контролю демографічного росту на спаду, зміні генетичного коду та популяції певних видів рослин та тварин, впливу на екологію, зміну ландшафтів та їхнього різноманіття. Таке моделювання допомагає в дослідженні становлення та занепаду стародавніх цивілізацій, з його допомогою досліджують еволюцію етноцентричної поведінки стародавнього світу, вимушене або добровільне кочівництво і міграцію груп, закономірності встановлення національної мови різних країн та народів, пізнавальне моделювання. Тобто таке моделювання добре себе проявляє і в атропологічній сфері. Зокрема створено різноманітні біомедичні додатки, включаючи моделювання 3D тканин молочної залози, дослідження різноманітних формутворень в людському тілі, вплив іонізуючого випромінювання на молочні стовбурові клітини, запалення та загалом імунну систему людини.

Моделі з використанням агентів для знаходження поведінки певних систем також використовуються для розробки та вдосконалення систем підтримки прийняття рішень, наприклад для діагностування при раку молочної залози. Такі моделі все повсимісно можуть використовуватися з метою моделювання фармакологічних рішень на ранніх стадіях і доклінічних дослідженнях, а також для допомоги в розробці лікарських засобів і дослідженні біологічних систем. Агентне моделювання застосовується для створення систем, що імітують поведінку організму людини або його окремих органів, де агентами виступають молекули.

Зокрема, такі агентно-орієнтовані моделі вже 30 років використовуються при рішенні різноманітних економічних та технологічних проблем. Такі рішення включають в себе імітацію бізнес-розуміння та сприйняття, економічні реакції клієнтів, роботу людей в колективі, оптимізацію логістичних рухів, моделювання поведінки споживачів, поведінку людей у віртуальному середовищі соціальних мереж та імітації роботи обслуговуючого персоналу для корпорацій, що мають велику кількість підлеглих працівників.

Агентне моделювання використовується для виявлення законів та тенденцій утворення заторів на дорогах та дослідженні потоку автомобілів вцілому.

Деякі застосування використовують агентне моделювання при навчанні самокерованих автомобілів при вирахуванні можливого маршруту та ситуацій на доргах, в тому числі і дорожно-транспортних пригод.

Останнім часом, агентні моделі були застосовані до різних сфер таких, як вивчення впливу місць публікації дослідників в галузі інформатики (журнали проти конференцій).

Також таке моделювання досліджує загальні соціальні явища. Агентні соціальні моделі застосовуються в дослідженнях та перевірці неперевіраних гіпотез. Такий підхід моделювання набуває ефективності у тому випадку,

якщо відсутні інші варіанти спостереження за агентами та їхніми діями. Часто це відбувається по причині того, що такий процес є довготривалим (наприклад, протягом створення нової мови, що займає кілька сотень років в реальному часі або небезпечним для реальних агентів. Іншою перевагою цього моделювання є те, що для того, щоб підтвердити теорію в симуляції, достатньо, щоб вона була представлена у формальній та логічній формі. Це призводить до більш ясного формулювання теорії.

Коли метою агентної моделі постає симуляція економічних та соціальних явищ (наприклад переміщення, провалів ринку, співпраці між групами або посилення конфлікту), то можна виявити такий процес, за якого модель, що була побудована на простих правилах, в кінці кінців показує складну динаміку. Таке явище пов'язється з концептом виникнення поведінки у суспільній науці.

#### 1.4 Проблеми агентного моделювання поведінки

Найбільш істотними проблемами, що виникають при вирішенні завдання агентного поведінки є:

- неможливість створення такого моделювання агента, що повністю охоплюватиме усі аспекти поведінки реального об'єкта в системах, що моделюють складні, не повністю вивчені організми та соціальні системи;
- в системах, що проводять передбачення на основі агентного моделювання проблемою є збільшення складності моделі із числом динамічно змінюваних параметрів у випадку, коли йдеться про точне передбачення;
- необхідність спрощення моделі по причині того, що технічні засоби, що використовуються не мають змоги опрацювати дані та надати бажаний результат за прийнятний відрізок часу;
- складність отримання статистики та результатів, а також їхня інтепретація;

— часта потреба в візуалізації процесів по причині необхідності контролю розвитку системи оператором;

— проблема переходу від логічно-дискретних подій прийняття рішень агентів до динамічних дій в середовищі, що розвивається в реальному часі.

### 1.5 Засоби реалізації систем агентного моделювання

У вирішенні таких проблем немає нічого такого, що відрізняло б від вирішення інших задач в засобах реалізації, оскільки основними відмінностями є зміст та дані, якими вона оперує. Тому для реалізації може бути будь-яка високорівнева мова програмування. В разі необхідності візуалізації процесів, можливо застосовувати як дво- так і тривимірну графіку будь-якої графічної бібліотеки або фреймворку.

Найбільше застосування знайшли в цій темі такі мови як Python, C++ та C#. Для першої існує достатня кількість налаштувань, бібліотек та фреймворків, що допомагають оптимізувати роботу програми та полегшити процес написання програмного забезпечення.

### 1.6 Опис існуючих платформ агентного моделювання

Однією з програм для розробки агентно-орієнтованих моделей є безкоштовне застосування NetLogo [3]. Розроблялася дана платформа в якості навчального інструменту, проте на сьогоднішній день її використовують зокрема і в дослідженнях.

За відгуками користувачів вона ідеально підходить для вивчення основ агентного моделювання.

В даному застосуванні є одиниця – агент, котрому можна задати набір правил, яким він буде слідувати. Наприклад, є два агенти – вівця та вовк. Кожен з них має наступні можливості: рухатися, їсти та давати потомство. Трава знаходиться по всьому полю, але для того щоб вівця її їла – вона має

витратити час на це і не рухатися, у той час як вовк їсть овець. В залежності від задання початкових даних, що виражаються в кількості початкових популяцій, можна спостерігати різні ситуації, що настають по проходженню певного часу. При налаштуванні занадто великої кількості вовків, вони з'їдають овець надто швидко доки ті не встигають розмножитися, при протилежних початкових даних вівці розможуються настільки швидко, що з'їдають усю траву вся екосистема помирає.

Є можливість досягти балансу екосистеми. На рисунку 1.1 вказано приклад роботи програми, де зліва показано середовище з домінуючими вовками, а зправа збалансоване.

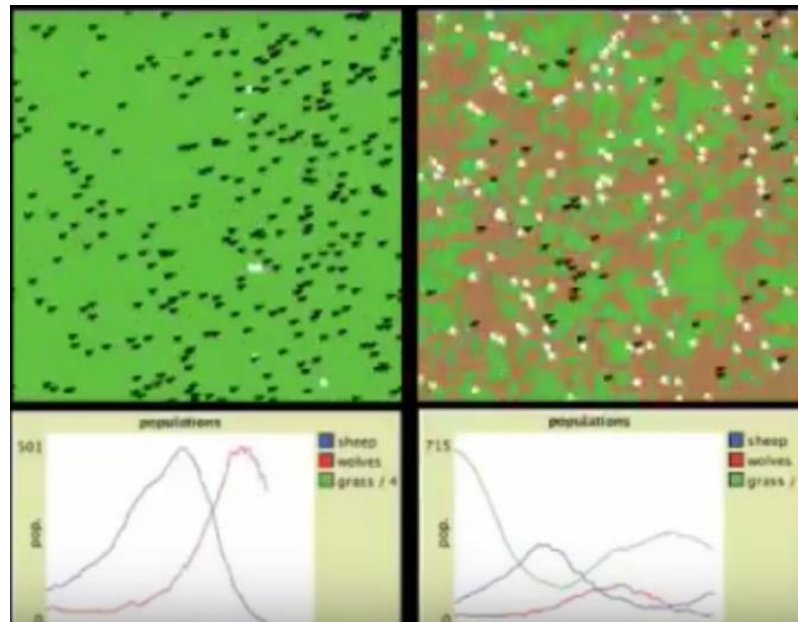


Рисунок 1.1– Приклад роботи NetLogo [3]

Інструментом для реалізації ширшого спектру в області агнетно-орієнтованого моделювання є програма Swarm, де використовується мова програмування Objective-C [4]. Програмувати в середовищі Swarm можна також і на Java.

## 1.7 Існуючі рішення агентного моделювання натовпу в екстремальних ситуаціях.

Проблема моделювання поведінки натовпу досліджується багатьма компаніями по причині того, що такі рішення доволі корисні для інженерів та архітекторів будівель для правильного розрахунку планів будівель та отримання інформації про майбутній рух людей у великих аренах, стадіонах, вокзалах, аеропортах.

Одною з найбільш відомих таких компаній є Pedestrian Dynamics, що мають одноіменне рішення для агентного моделювання поведінки натовпу [5]. Застосування є комплексним програмним забезпеченням моделювання натовпу. Воно призначене для створення і відтворення великих моделей пішохідного моделювання в складних інфраструктурах. Pedestrian Dynamics може бути використане для оцінки ефективності та безпеки середовища в кожній фазі життєвого циклу моделювання системи руху агентів від проектування до операції.

Програма пропонує швидко опановуване середовище моделювання побудови моделі, що економить час і витрати на розробку. Користувач має виконати кілька кроків, які необхідно моделювати пішохідні потоки в найскладніших операціях. Є гнучким, надійним і простим у використанні. Широко використовується в багатьох великомасштабних проектах для імітації натовпу потоків в найбільш критичних умовах інфраструктури, включаючи аеропорти, термінали громадського транспорту та міського планування.

Клієнти використовують вищезгадане програмне забезпечення для моделювання для імітації великомасштабних логістичних систем і інфраструктур, таких як системи обробки багажу, контейнерних терміналів, залізничних станцій, складальних ліній і футбольних стадіонів. Програмне забезпечення, що здійснює моделювання дозволяє користувачеві впоратися з пливом часу, витратами, ресурсами, надійністю, безпекою і стабільністю.

					IA61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		16

На рисунку 1.2 вказано приклад роботи даного застосування, а саме евакуації стадіону. Звісно, при правильному налаштуванні, це програмне забезпечення дозволяє відстежувати потоки людей в різних випадках, в тому числі і евакуації, проте основним його мінусом, на мою думку є те, що під час цього процесу не враховується психоемоційний стан натовпу та моделювання не дасть повних результатів у випадку виникнення паніки у натовпі під час вибуху, пожежі чи іншої екстремальної події. При таких подіях не слід очікувати, що натовп буде виходити спокійно та в черзі, як вказано на рисунку.



Рисунок 1.2– Евакуація стадіону в Pedestrian Dynamics

Ще однією такою розробкою є робота Олівера Жиманежика, Тома Бакета та Патріка Дікінсона, що таке імітує поведінку людей в приміщенні аеропорту, досліджує пасажиропотік і моделює рух людей в цілому [6]. Вони адаптували популярний ігровий рушій до представлених так, як його вибір значно спростив програмування та взаємодії агентів, а особливо його графічне преставлення.



Тривимірні агенти рухаються у двовимірному просторі та піддаються фізичним законам динаміки, тобто керують силами свого руху та можуть взаємодіяти з іншими агентами. Середовище представлене розмірами 200 на 150 метрів, і розділене на клітини 5 на 5 метрів, поставлено термінали, колони, агенти, дошки оголошень та інші фізичні об'єкти відповідно до плану реальної будівлі, а саме аеропорту Гатвіг. Це зробило розробку більш наглядною та зрозумілою.

В даній програмі агенти фактично пересуваються від клітини до клітини по своєму заданому маршруту з імітацією вільного руху, але фактично вони рухаються обираючи наступну клітину. Є шар навігації агентів, за яким вони обирають напрямки.

Застосування дозволяє рухати камеру в будь-якому напрямку та оглядати системи з різних ракурсів. На рисунку 1.3 вказано заповнення терміналу аеропорту агентами.

Проблемою даного застосування є та ж недостача моделювання екстремальних подій.

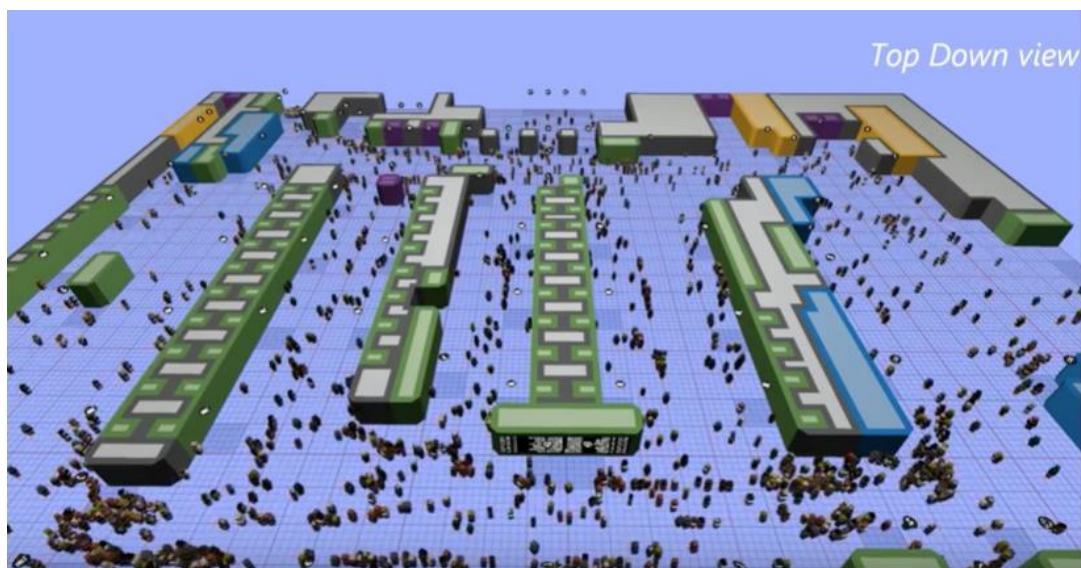


Рисунок 1.3 – Рух в моделі аеропорту в науковій роботі



## 2 АНАЛІЗ ІСНУЮЧИХ СТРУКТУР НЕЙРОННИХ МЕРЕЖ ТА МЕТОДІВ НАВЧАННЯ

### 2.1 Штучні нейронні мережі та способи їх побудови

Штучна нейронна мережа (ІНС) є парадигмою обробки інформації, до створення якої причетна найбільшим чином біологічні нервові системи, такі як мозок, що здатен обробляти інформацію базуючись на зв'язках нейронів. Ключовим елементом цієї парадигми є недосліджена, оригінальна структури системи обробки інформації. Ця система складається з великого числа тісно взаємопов'язаних елементів обробки (нейронів), що працюють разом, щоб вирішити специфічні проблеми.

Штучні нейронні мережі, як і люди, вчаться на прикладі. ШНС налаштовується для конкретної задачі, такої як розпізнавання образів або класифікація даних, та досягає результатів виконання через процес навчання. Навчання в біологічних системах включають коригування синаптичних зв'язків, які існують між нейронами, отже для штучних нейронних мереж це також працює.

Нейронні мережі, з їхньою корисною в багатьох випадках здатністю виокремлювати інформацію зі складної або неточної інформації, можуть бути використані для отримання моделей і виявлення тенденцій, які занадто складні, щоб бути поміченими людиною або іншими методами програмних рішень. Правильно навчену нейронну мережу можна розглядати як «експерта» в категорії інформації, яку вона аналізувала. Такий експерт може бути використаний для надання прогнозів, моделювати нові ситуації, що є її компетенції і прогнозувати події для різних випадків та питань. Нейронні мережі мають такі переваги порівняно з іншими підходами:

— адаптивне навчання: здатність навчитися виконувати завдання, засновані на даних, наданих для навчання або початковому досвіді;

- самоорганізація: ШНМ може створити свою власну організацію або подання інформації, яку він отримує під час навчання;
- у реальному масштабі часу роботи: обчислення можуть здійснюватися паралельно, зокрема розробляються спеціальні процесори та інші спеціальні апаратні пристрої, які дозволяють скористатися цією можливістю;
- відмовостійкість через кодування з надмірністю інформації: часткове руйнування мережі призводить до відповідного зниження продуктивності. Тим не менш, деякі мережеві функції можуть бути збережені навіть при наявності великого пошкодження мережі. При отриманні відповідної можливості, вона зможе заново навчитися та заповнити прогалини.

Нейронні мережі використовують інший підхід до вирішення проблем, ніж звичайні комп'ютерні рішення. Останні використовують алгоритмічний підхід тобто комп'ютер має набір інструкцій для того, щоб вирішити проблему. Без таких конкретних кроків, котрим комп'ютер повинен слідувати, очевидно, він не може вирішити цю проблему. Це обмежує можливості вирішення звичайними комп'ютерів проблем, тими, які розробник вже розуміє і знає, як вирішити.

Нейронні мережі виконують інформаційні процеси подібним чином до того, як це робить людський мозок. Мережа складається з великого числа тісно взаємопов'язаних елементів обробки (нейронів), що працюють паралельно, щоб вирішити конкретну проблему. Нейронні мережі навчаються на прикладі. Вони не можуть бути запрограмовані для виконання конкретного завдання, а приклади повинні бути ретельно відібрані. В іншому випадку час навчання витратиться даремно, а можливо, мережа стане функціонувати неправильно.

Недолік полягає в тому, що, оскільки мережа сама знаходить рішення проблеми, її дії можуть бути непередбачуваним.

## 2.2 Огляд штучного нейрона

					IA61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		20

Штучний нейрон (базовий процесорний елемент) — є основою будь-якої штучної нейронної мережі.

Нейрони являють собою відносно прості, однотипні елементи, що імітують роботу нейронів мозку. Кожен нейрон характеризується своїм поточним станом (збуджений або загальмований) за аналогією з нервовими клітинами головного мозку.

Штучний нейрон, також як і його природний прототип, має групу синапсів (входів), які з'єднані з виходами інших нейронів, а також аксон — вихідну зв'язок даного нейрона, звідки сигнал збудження або гальмування надходить на синапси інших нейронів. Він складається з двох елементів — зваженого суматора і нелінійного перетворювача. Будову штучного нейрону вказано на рисунку 2.1.

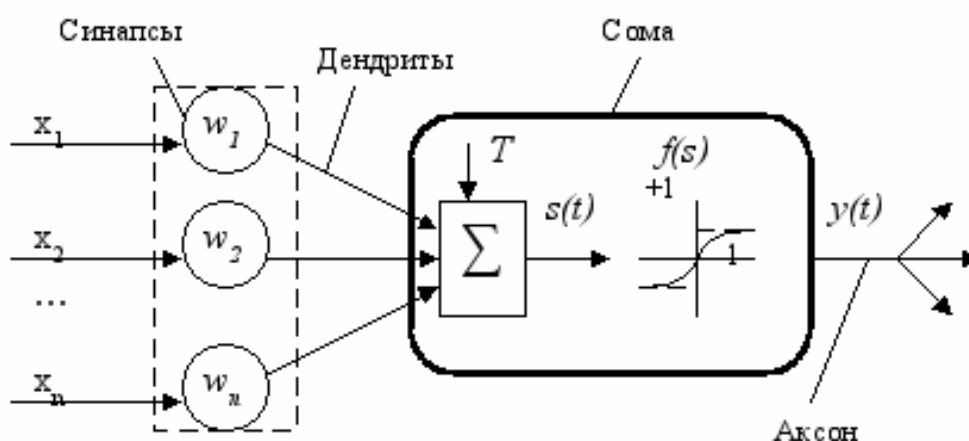


Рисунок 2.1– Схематичне зображення штучного нейрона

На вхід штучного нейрона надходить деяка безліч сигналів, кожен з яких є виходом іншого нейрона. Кожен вхід множиться на відповідну вагу, аналогічний синаптичній силі, і всі твори підсумовуються, визначаючи рівень активації нейрона.

Зважений суматор виконує сумування за формулою:

$$S_{jl} = \sum_{i=1}^n X_i W_{ijl} + W_{0jl} , \quad (2.1)$$

де,  $i$  – номер входу нейрона;  $j$  – номер нейрона в шарі;  $l$  – номер шару,  $X_i$  – вхідні сигнали, сукупність всіх вхідних сигналів нейрона, утворюючих вектор  $\mathbf{X}$ .  $W_i$  – вагові коефіцієнти, сукупність вагових коефіцієнтів, що утворюють вектор ваг  $\mathbf{W}$ .  $W_0$  – вага, що моделює пороговий рівень нейрона.

$\mathbf{S}$  – зважена сума вхідних сигналів, значення передається на нелінійний елемент. Її формула є наступною:

$$\mathbf{S} = \mathbf{X}^T \mathbf{W}, \quad (2.2)$$

де  $\mathbf{X}$  – це вектор вхідних сигналів нейрона, що подаються на вход з вагою  $W_0$ ;  $\mathbf{W}$  – вектор ваг нейрона, що включає  $\mathbf{W}_0$ .

Нелінійний елемент перетворює вихід суматора за формулою:

$$\mathbf{Y}_{jl} = f(\mathbf{S}_{jl}) \quad (2.3)$$

де  $f$  – це функція активації, яка підбирається специфікою розв'язуваної задачі, зручністю реалізації нейронної мережі і алгоритмом навчання.

Серед найпростіших типів штучних нейронів найбільш відомий «Adaline», створений Б. Уїдроу, для якого функція  $F(x)$  – лінійна, а вихід визначається співвідношенням:

$$q = \sum_{j=1}^n \mathbf{w}_j \mathbf{r}_j + w_0 \cdot 1, \quad j = \overline{0, n}, \quad (2.4)$$

У якій  $\mathbf{w}$  – вагові коефіцієнти,  $\mathbf{r}$  – входи

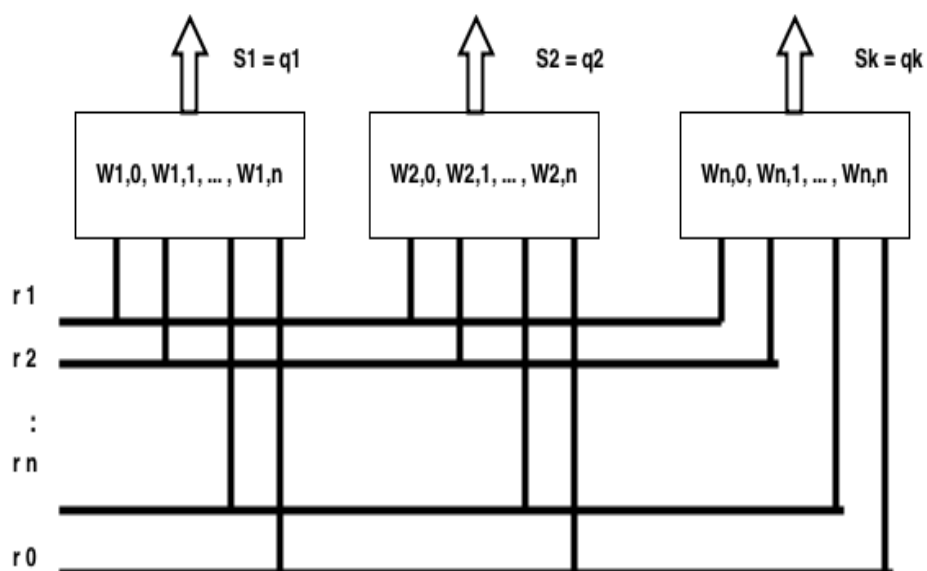


Рисунок 2.2 – Схематичне уявлення одношарової лінійної штучної нейронної мережі [7]

Найпростіша одношарова лінійна нейронна складається з  $K$  нейронів типу «adaline» (рисунок 2.2) і називається «madaline». Вона має  $K$  виходів і входів і може бути записана у векторно-матричній формі так:

$$q = \mathbf{W}^{(K)} r_1 + w_0 r_0 = \mathbf{W}^{(K)} \mathbf{r}, \quad (2.5)$$

де  $\mathbf{q} = \text{col}(q_1, \dots, q_i, \dots, q_k)$  — вектор виходу;  $\mathbf{r}_1 = \text{col}(r_1, \dots, r_j, \dots, r_n)$  — вектор входу; з урахуванням входу ініціалізації  $r_0$ , однакового для всіх адалін, вектор входу  $\mathbf{r}^T = (r_1^T : r_0)$ ;  $\mathbf{W}^{(K)} = \| w_{ij} \|, i = \overline{1, K}, j = \overline{0, n}$  — матриця вагових коефіцієнтів;  $\mathbf{W}_1^{(K)} = \| W_{ij} \|, i = \overline{1, K}; j = \overline{1, n}$  — матриця, одержувана з матриці  $\mathbf{W}^{(K)}$  видаленням останнього стовпчика — вектора коефіцієнтів ініціюючих зв'язків  $w_0 = \text{col}(w_{1,0}, \dots, w_{k,0})$ .

У відповідності з формулою (2.5) лінійна багатошарова мережа типу «madaline» здійснює відображення  $\mathbf{r}_1 \in R^n$  в  $\mathbf{q} \in R$  [7].

## 2.3 Архітектура штучних нейронних мереж

Штучні нейронні мережі конструюються з основного блоку - штучного нейрону. Інша властивість нейромереж є величезна кількість зв'язків, які пов'язують окремі нейрони між собою.

Біологічні нейромережі складаються з мікроскопічних компонентів й існують у трьохвимірному просторі і здатні до різноманітних з'єднань. Проте для того, аби реалізувати штучну мережу, присутні фізичні обмеження.

Об'єднавшись у мережу, штучні нейрони створюють систему обробки інформації, що забезпечує чітку адаптацію моделі до різноманітних змін з боку навколишнього середовища. Перетворення вхідного вектора сигналів у вихідний відбувається під час функціонування мережі. Кожен вид перетворення відрізняється архітектурою нейромережі, характеристикою нейронних елементів, способами керування та синхронізацією інформаційних потоків.

За архітектурою зв'язків ІНС можуть бути згруповані в два класи (рисунок 2.3): мережі прямого поширення, і рекурентні мережі (із зворотними зв'язками).

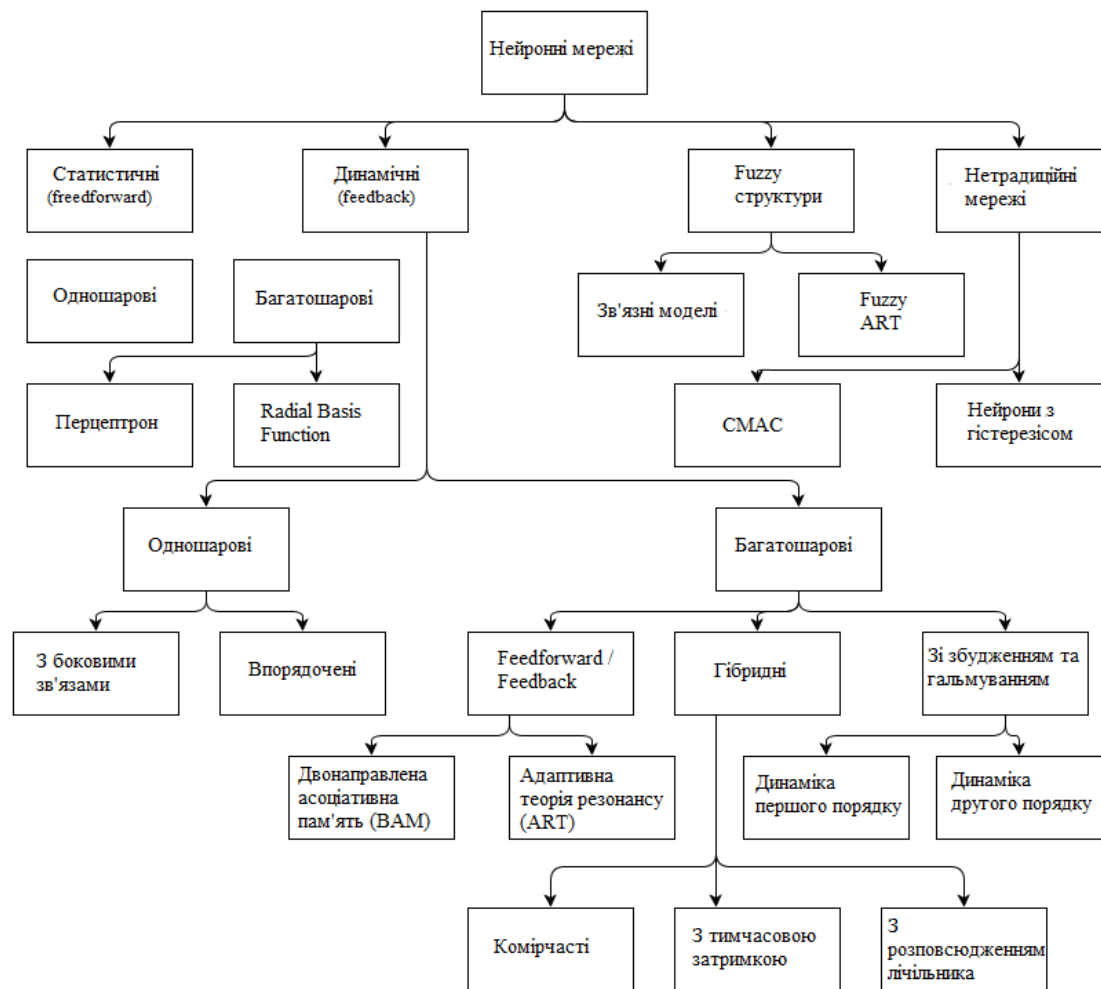


Рисунок 2.3 – Класифікація штучних нейронних мереж

У мережах прямого поширення сигнал по мережі проходить тільки в одному напрямку: від входу до виходу. Сукупність нейронів, об'єднаних в один шар, називають одношарової нейромережею.

Об'єднання одношарових нейромереж в кілька шарів визначає багатошарову нейромережу.

### 2.3.1 Одношаровий персептрон

На вхід надходить тільки двійковий сигнал, тобто або 0 або 1. Елемент  $\Sigma$  (суматор) складає вхідні сигнали, зважені щодо відповідних синапсів нейрона. Якщо ця сума більше заданого порогового значення, вихід дорівнює 1, якщо менше — нулю. Оскільки вихідний сигнал приймає значення

«0» або «1», тому персептрон — це нейрон бінарного типу. При функціонуванні персептрона розрізняють два режими його роботи:

- режим навчання;
- режим функціонування.

Модель персептрона, представлена на рисунку 2.4, складається з одного шару (тобто кількість шарів нейронів між входом і виходом дорівнює одиниці) штучних нейронів, з'єднаних за допомогою вагових коефіцієнтів з безліччю входів.

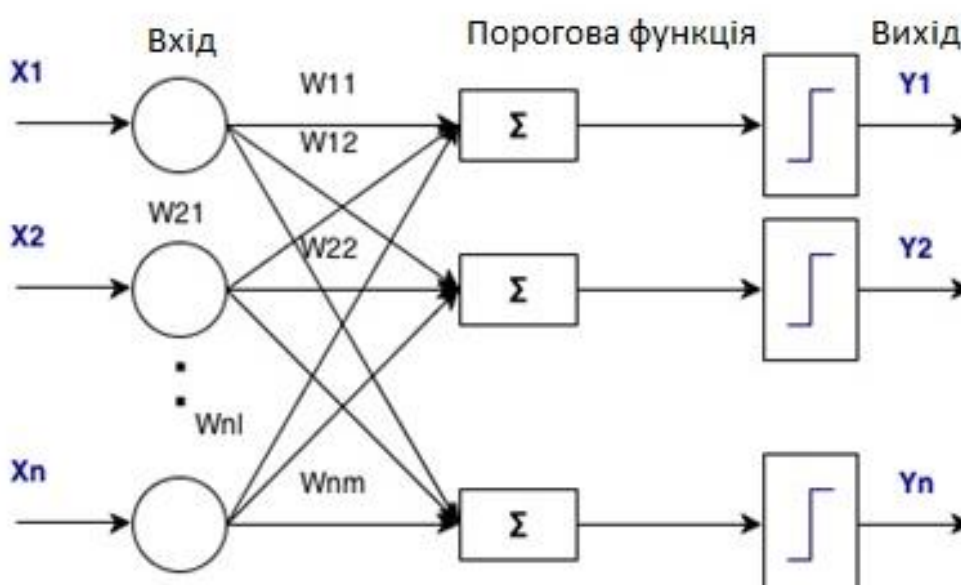


Рисунок 2.4 – Структура одношарового персептрона

### 2.3.2 Багатшаровий персептрон

Багатшарова мережа складається з нейронів, які розташовані на різних рівнях. Крім вхідного і вихідного шарів є як мінімум ще один шар — прихований. При завданні функції активації сигмоїдального типу матимемо багатшаровий персептрон (рисунки 2.5).



У багатошаровому персептроні кожен нейрон на даному рівні ієрархії приймає і обробляє сигнали від кожного нейрона нижчого рівня. Прихований шар нейронів дозволяє мережі навчатися вирішення складних завдань, послідовно отримуючи найбільш важливі ознаки з вхідного шару.

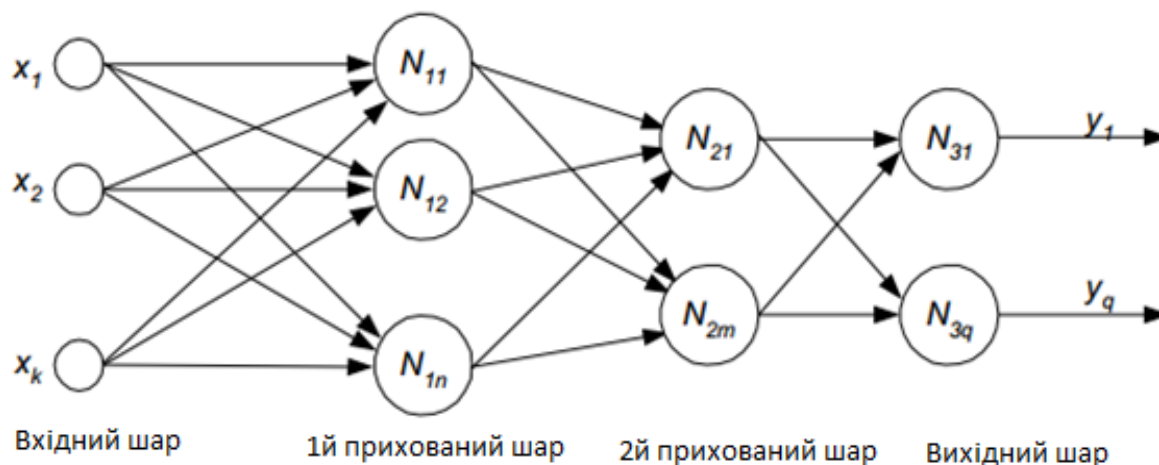


Рисунок 2.5 – Структура багатошарового персептрона

Багатошарові персептрони виявилися ефективними для вирішення різноманітних завдань управління.

### 2.3.3 Рекуррентні мережі

Нейронні мережі, що мають зворотний зв'язок, називаються «рекуррентними мережами».

Кожен крок мережі називається ітерацією. Рекуррентна мережа може складатися з єдиного шару нейронів, кожен з яких спрямовує свій вихідний сигнал на входи всіх інших нейронів шару.

Наприклад, мережа Хопфілда, представлена на рисунку 2.6 не має прихованих нейронів.

Варто відзначити, що на малюнку видно, що відсутні зворотні зв'язки нейронів з самими собою.

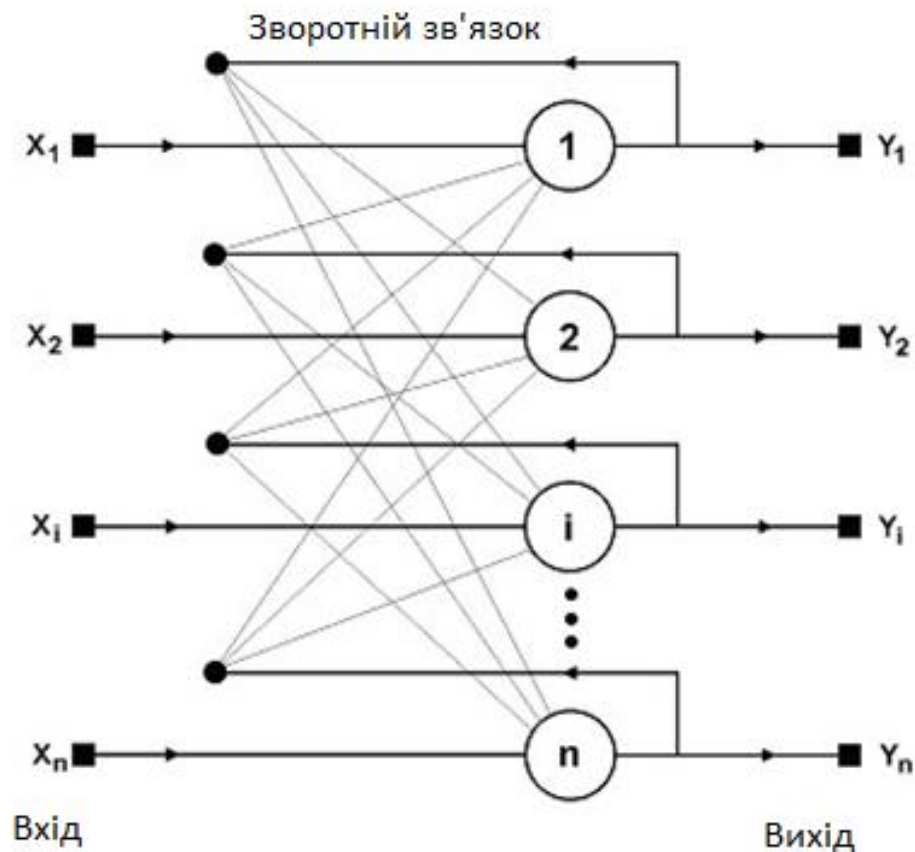


Рисунок 2.6 – Структура мережі Хопфілда

На малюнку нижче (рисунок 2.7) представлений інший клас рекурентних мереж з прихованими шарами нейронів. У цьому випадку, зворотні зв'язки виходять, не просто тільки з вихідних нейронів, а ще й з прихованих нейронів.

Наявність зворотного зв'язку в мережі, показаної на рисунку 2.7. Вона має безпосередній вплив не тільки на продуктивність, а й до того ж, здатність мережі до навчання.

Слід зазначити, що зворотний зв'язок має на увазі використання елементів одиначної затримки (позначені як  $z-1$ ), що призводить до нелінійного динамічного поведінки, якщо, звичайно, в мережі містяться нелінійні нейрони.

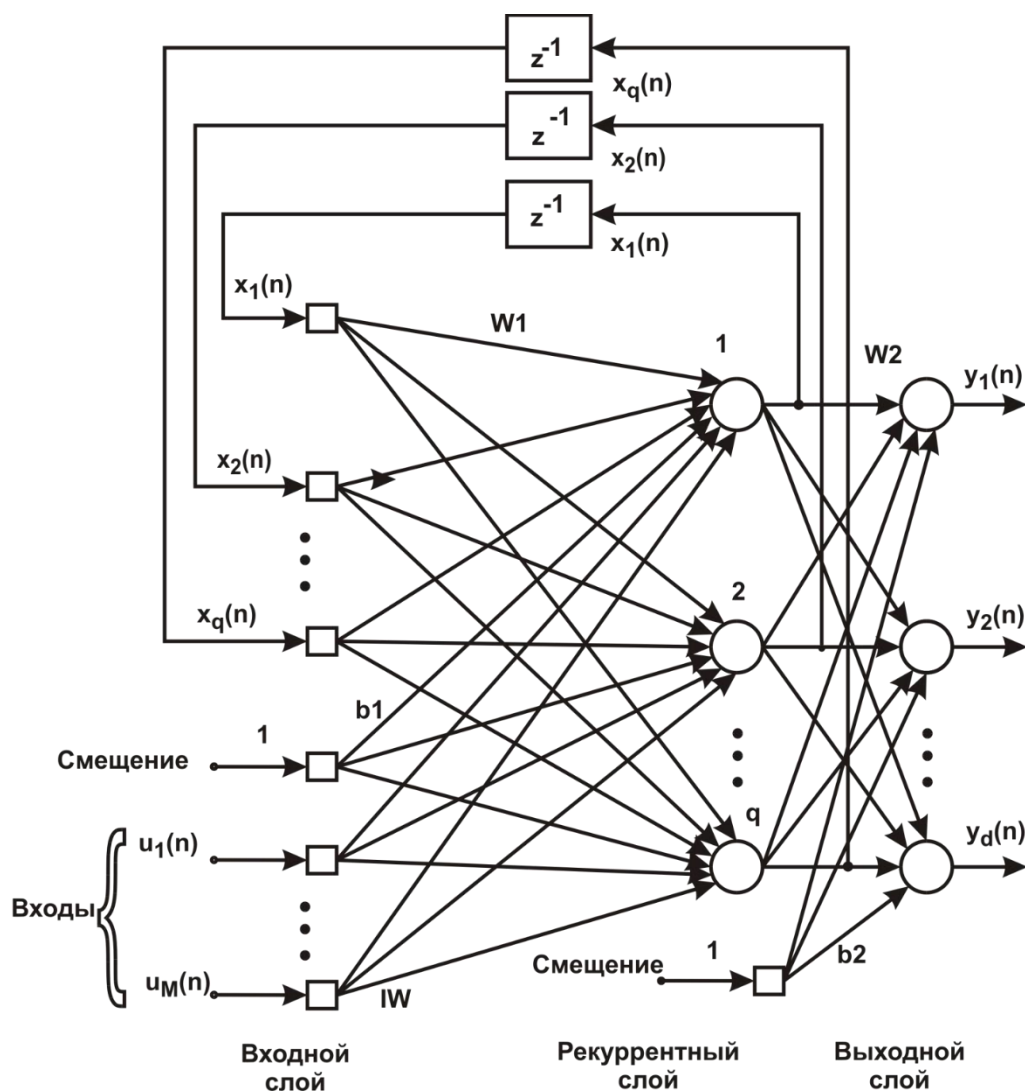


Рисунок 2.7 – Структура рекуррентной мережі з прихованими нейронами.

## 2.4 Навчання нейронних мереж

Процес навчання нейронних мереж є налаштування архітектури мережі, а також її вагових коефіцієнтів для ефективного вирішення поставленого завдання. Мережа повинна налаштувати ваги по навчальним прикладам. Саме ця особливість навчання на прикладах і виділяє нейронні. Існує два класи навчання:

- детермінований метод навчання;
- стохастичний метод навчання.

Детермінований метод — це метод, при якому параметри мережі змінюються по кроках (ітераціям), ґрунтуючись на поточних параметрах входів і бажаних виходів. Найбільш популярним прикладом детермінованого методу є алгоритм «зворотного поширення помилки»

При стохастичному методі параметри мережі змінюються випадковим чином. Але зберігаються тільки ті зміни, які привели до поліпшень.

#### 2.4.1 Навчання з учителем та без

Є кілька підходів до навчання нейронних мереж. Один з них — це навчання з учителем. На рисунку 2.8 показана блокова діаграма, що ілюструє цю форму навчання.

Фактично участь вчителя виражається як наявність інформації про навколишнє середовище, що представлено у вигляді пар вхід-вихід. При цьому саме середовище невідоме ненавченій нейронній мережі. Тепер припустимо, що вчителю і ненавченій мережі подається навчальний вектор з навколишнього середовища.

На основі вбудованих знань учитель формує і передає нейронній мережі бажану відповідь, яка відповідає даному вхідному вектору. Такий результат являє собою оптимальні дії, які повинна виконати нейронна мережа. Параметри мережі коригуються з урахуванням навчального вектора і величини помилки. Величина помилки — це різниця між бажаною відповіддю і поточною відповіддю нейронної мережі.

Коригування параметрів виконується крок за кроком з метою імітації нейронною мережею поведінки вчителя. Цей процес має бути оптимальним в деякому статистичному сенсі. Таким чином, в процесі навчання знання

вчителя надаються мережі в повному обсязі. Після закінчення навчання вчителя можна відключити і дозволити нейронній мережі працювати з середовищем самостійно.

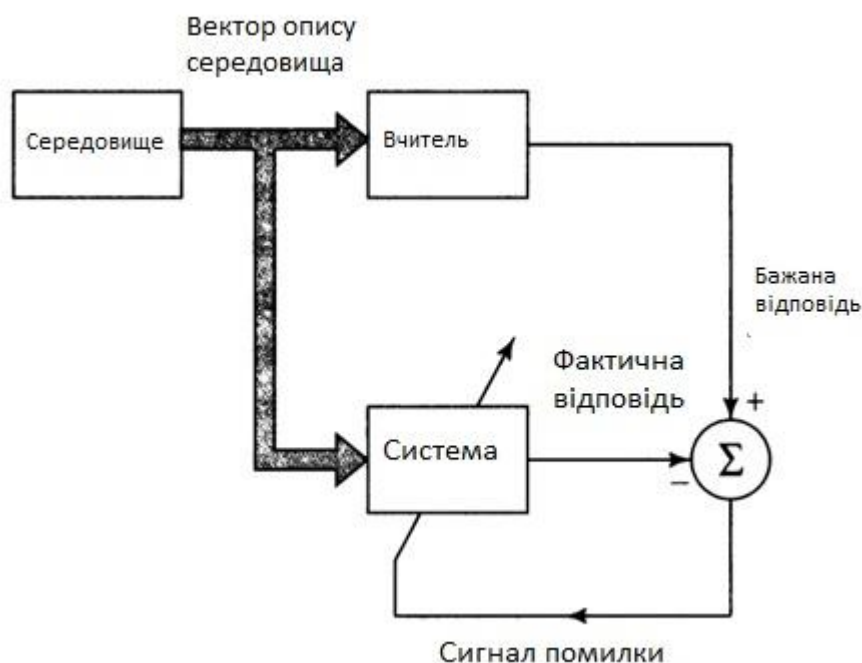


Рисунок 2.8 – Схема навчання мережі з учителем.

Описаний вище процес навчання відбувається під керівництвом вчителя. Інший спосіб навчання передбачає відсутність вчителя, а отже система має сама отримати правильні відповіді, або повинна мати певний інструмент, що забезпечує вибір кращого рішення. Одним з таких є генетичний алгоритм, що обирає найкращі рішення та породжує з них ще кращі.

#### 2.4.2 Генетичний алгоритм навчання

Генетичні алгоритми є адаптивним алгоритмом пошуку евристичного на основі еволюційних ідей природного відбору і генетики. Таким чином, вони являють собою інтелектуальну експлуатацію випадкового пошуку, що використовується для вирішення завдань оптимізації. Незважаючи на те що

він є рандомізованим, його робота ні в якому разі не випадкова, замість цього алгоритм використовує інформацію з історії роботи системи для прямого пошуку найкращий даних та підвищення продуктивності. Основні методи генетичного алгоритму призначені для моделювання процесів в природних системах, необхідних для розвитку, особливо ті, що слідують принципам, викладеним Чарльзом Дарвіном з «виживання найбільш пристосованих». Так як в природі конкуренція серед осіб за мізерні ресурси показує в результаті домівання найбільш пристосованих особин над слабшими.

Генетичний алгоритм — це самонавчальний алгоритм, який застосовується в задачах оптимізації, і в основному там, де не завжди можливо сформулювати завдання у вигляді, придатному для більш швидких алгоритмів локальної оптимізації (наприклад, в оптимізації штучних нейронних мереж). Сам алгоритм є ітеративним і він заснований на принципах генетичних процесів біологічних організмів (схрещування). Шляхом перебору і відбору знаходиться необхідна комбінація.

Система, основана на цьому алгоритмі краще, ніж звичайний штучний інтелект в тому, що вона є більш надійною.

На відміну від старих систем штучного інтелекту, вони не виходять зі строю легко, навіть якщо вхідні дані змінилися незначно, або при наявності помірного шуму. Крім того, при пошуку у багатомірному просторі даних, мультимодальні в просторі станів, або n-мірну поверхню, генетичний алгоритм може запропонувати значні переваги в порівнянні з більш типовим пошуком методів оптимізації.

Схрещування найбільш пристосованих особин призводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. В кінцевому підсумку, популяція буде сходиться до оптимального рішення задачі.

Алгоритм ділиться на три етапи та один додатковий:

- схрещування;
- селекція;

- формування нового покоління;
- мутація.

Якщо результат нас не влаштовує, ці кроки повторюються до тих пір, поки результат нас не почне задовольняти або станеться одне з нижче перерахованих умов:

- кількість поколінь (циклів) досягне заздалегідь обраного максимуму;
- вичерпаний час на мутацію.

Створення нової популяції. На цьому кроці створюється початкова популяція, яка може виявитися не задовольняє, але є ймовірність, що алгоритм зможе виправити цю проблему. Головне, щоб вони були пристосовані до розмноження. Розмноження. Для отримання нащадка потрібно два батька. Нащадок повинна успадкувати від батьків їх риси. При цьому розмножуються все, а не тільки ті, що вижили. Мутації. Мутації схожі з розмноженням, з мутантів вибирають певну кількість особин і змінюють їх відповідно до заздалегідь визначеними операціями. Відбір. З популяції вибирається частка тих, хто піде далі. Частка тих, хто вижив після відбору визначається параметрами, а решта особини повинні загинути.

На рисунок 2.9 - 2.10 вказано приклади мітозу та мутації. Принцип роботи генетичного алгоритму наведено в кресленику ІА.61.140БАК.001 Д2.

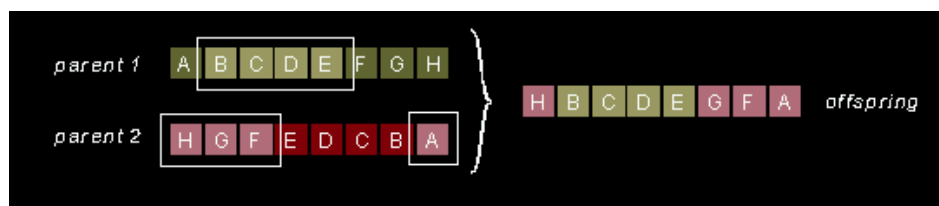


Рисунок 2.9 – Приклад мітозу генетичного алгоритму



## Рисунок 2.10 – Приклад мутації генетичного алгоритму

Генетичний алгоритм є досить потужним засобом і може з успіхом застосовуватися для широкого класу прикладних завдань, включаючи ті, які важко, а іноді і зовсім неможливо, вирішити іншими методами. Однак, генетичний алгоритм, як і інші методи еволюційних обчислень, не гарантує виявлення глобального рішення за поліноміальний час, не гарантує і того, що глобальне рішення буде знайдено, але вони гарні для пошуку «досить хорошого» рішення задачі «досить швидко». Головною ж перевагою генетичного алгоритму є те, що вони можуть застосовуватися навіть на складних завданнях, там, де не існує ніяких спеціальних методів.

### 2.5 Застосування генетичного алгоритму та штучних нейронних мереж в розробках з моделювання поведінки об'єктів

Створення агентних систем моделювання, що імітують повністю або частково реальні системи, в яких поведінка одиничних об'єктів визначає поведінку усієї системи в цілому є досить популярною проблемою. Першою досить відомою з таких систем була математична гра «Життя», запропонована Джоном Конвеєм [8].

Система представляє собою клітковий автомат, що має чіткі правила середовища та агентів у ньому. Територією є розмічена площа, клітки якої можуть знаходитися в одному з двох станів: бути «живими» або «мертвими». На початку гри живі клітини розподіляються рівномірно по середовищу, а наступне покоління розраховується за правилами: в порожній клітці, поруч з якою рівно три живі клітини, зароджується життя; якщо у живої клітини є дві або три живі сусідки, то ця клітина продовжує жити; в іншому випадку, якщо сусідів менше двох або більше трьох, клітина вмирає («від самотності» або «від перенаселеності»). Гра завершується якщо на

					ІА61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		34



полі не залишиться жодної «живої» клітини, конфігурація утворення клітин зациклюється або лишається незмінною, тобто система не може далі розвиватися. Так, з ростом обчислювальної потужності дослідження поведінки живих організмів за допомогою нейронних мереж набуває популярності, будуються все більші системи для вирішення цих проблем.

Одною з таких розробок є «Штучне життя», що імітує поведінку мікроорганізмів та має цілю дослідження оптимальної поведінки великої агентів у пошуках їжі в замкнутому середовищі [9]. Для досягнення такої застосовано нейронну мережу, що навчається за генетичним алгоритмом.

Дослідження показали, що при побудові певної топології мережі та настройці цілей генетичного алгоритму система досить швидко надає прийнятні та цікаві результати. Рухаючись спочатку хаотично, згодом агенти будують ефективну модель поведінки та можуть виживати на протязі часу, що в сотні разів перевищує початковий, а при додаванні нових характеристик формують колонії та конкурують між собою.

Схожі дослідження проводяться з різними середовищами та можливостями агентів також із застосуванням генетичного алгоритму так як він підходить для навчання таких систем [10]. Проте розробки, наведені вище досліджують лише поведінку найпростіших та не вирішують ряду проблем з моделювання та дослідження таких систем, що імітують, наприклад, людську поведінку, тому було вирішено запропонувати таку систему, що дозволить навчити агентів утворювати систему, що моделює поведінку, близьку до поведінки панікуючого натовпу.

### 3 РОЗРОБКА МЕТОДИКИ ТА СИСТЕМИ МОДЕЛЮВАННЯ ПОВЕДІНКИ НАТОВПІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ ТА ГЕНЕТИЧНОГО АЛГОРИТМУ

#### 3.1 Розробка системи, що керує модельованими об'єктами

Для моделювання поведінки натовпів найбільш наочним та ефективним на мою думку є підхід повного моделювання та регулювання моделей в реальному часі експериментально. Таким чином, було вирішено створювати систему наступним шляхом:

У 3-вимірній моделі великого приміщення з проходами, завадами, стінами та виходами знаходитимуться об'єкти у вигляді кубів, що стоять або рухаються процедурно навмання.

У певний момент деякі частини кімнати будуть займатися полум'ям. Перед кубічними об'єктами стоятиме завдання покинути приміщення при цьому уникаючи вогню та не створюючи затори на виходах із приміщення, вони мають це зробити базуючись на даних про попередні події з минулих ітерацій, так як поведінка та результат дій об'єктів записуватиметься до бази даних.

Тобто очікується, що спочатку, при перших запусках цього процесу об'єкти нездатні будуть покинути кімнату максимально ефективно для себе та оточуючих, так як у них буде недостатньо даних про те як не уникати вогню, не створювати натовпи та інше, проте на пізніх ітераціях вони зможуть організовуватися в «чергу» та не перешкоджати один одному з метою збереження більшої кількості об'єктів від вогню або оберуть егоїстичну стратегію та намагатимуться врятуватися якомога швидше, не враховуючи інші об'єкти навколо.

Отже, метою навчання об'єктів є за допомогою генетичного алгоритму будувати свою поведінку виходу з занятого приміщення, властивості та наповнення якого змінюється від ітерації до ітерації.

Агенти матимуть певні стратегії поведінки, що будуть вмикатися та реагувати при певній події. Ці стратегії обумовлені обробленою інформацією про систему, подію чи роботу організму.

При застосуванні генетичного алгоритму буде виведено натовп агентів з різним набором таких стратегій, який сумарно даватиме результати, вказані в функції пристосованості.

Тобто отримана система поводитиме себе згідно з потрібними результатами, вказаними оператором. Так, її навчання може бути націленим як на отримання найефективнішої популяції, що залишає будівлю якомога швидше з найменшою кількістю жертв, так і на досягнення інших показників з метою підвищення реалістичності її поведінки.

Можливо буде змодельовати певну екстремальну ситуацію, що вже сталася в минулому в ідентичному середовищі та запустити навчання до досягнення схожої конфігурації поведінки агентів.

Таким чином, такі навчені моделі можуть вважатися достатніми для використання в інших середовищах.

### 3.2 Огляд стратегії поведінки модельованого об'єкта

Так як інструментом моделювання було обрано агентний підхід, то було вирішено створити таку систему, що охоплюватиме лише індивідуальну поведінку кожного агента, але при цьому в повній мірі забезпечуватиме побудову цієї поведінки. Як вказано вище, реальний натовп має тенденцію до утворення масової свідомості та прийняття масових рішень. Таке спілкування також мало бути враховане в самих поведінках та системі їхнього підбору.

Було вирішено побудувати взаємодію агента з навколишнім середовищем та іншими агентами базуючись на певних подіях та реакціях агентів на них. Тобто внесено набір певних тактичних та стратегічних рішень як реакцію на

події та розподіл цих подій, на ті, що впливають на курс руху об'єкта та вирішуються правильним підбором стратегії, що реагуватиме на такі події та ті події, що вимагають від об'єкта короткочасної реакції відповідно по обраної тактичної поведінки, як то обхід вогню або перехід, одноразова реакція на дії інших агентів у момент коли взаємодія з ними доцільна та можлива.

Звісно, можливо будувати стратегії так, щоб вони охоплювали також деякі тактичні моделі поведінки, при цьому спрощуючи побудову доречного набору поведінки агентом.

Прикладом такої є стратегія обходу вогню за допомогою нейронної мережі, що описана нижче, так я вона показує і тактику обходу вогню та інших об'єктів, а також частково охоплює стратегію руху до виходу та вибору найбезпечнішого з них.

### 3.2.1 Розробка початково необхідного переліку можливих типів поведінки до стратегій та тактик

Список стратегій та тактик поведінки можна розширяти до тих пір, доки агенти не матимуть такий набір поведінок, що ідентичний людському. Проте, такий процес розробки та дослідження може зайняти надто велику кількість часу, тому необхідно виділити перелік мінімально необхідних стратегій для розробляємої системи.

Так як в даній роботі розглядається тільки ситуація утворення пожежі в приміщенні закритого типу та реакції натовпу на неї, то найнеобхіднішими тактичними подіями виділено: наближення полум'я, факт штовхання з боку інших об'єктів. Також стратегічні події: утворення вогню, агент почув чужі крики паніки, наближення до виходу з приміщення.

Даний список подій є найменш повним, проте достанім для забезпечення моделювання агентної системи, що описана вище. Існує можливість доповнювати список в залежності від ситуації, що моделюється.

Отже, було створено початковий список стратегій та тактик відповідно до мінімального переліку подій. Для кожної події запропоновано найоптимальніший варіант розвитку подій, близький до поведінки реального натовпу. Таким чином, в результаті запрограмовано від двох можливих стратегій для забезпечення варіативності та можливості дослідження кращої з них.

Таблиця 3.1 – Приклад набору стратегій агента

Тип руху	Виявлення вогню	Почув панічні крики	Побачив натовп перед собою	Реакція на штовхання
Рух до цілі незважаючи ні на що	рухатися прямо до найближчого головного виходу, незважаючи ні на що	почати стратегію в виявленні вогню	обрати інший шлях виходу	ігнорувати
Перед перепорою змінювати вектор руху на 30°	рухатися до найближчого виходу, що помітив, почати крик	ігнорувати крики	проходити ігноруючи натовп	штовхнути у відповідь
Рух на основі стратегії нейронної мережі 1	рухатися до найближчого виходу, якщо той не зайнявся полум'ям		рішення на основі стратегії нейронної мережі	

Агенти з'являються в середовищі утворюючи своєю кількістю натовп і рухаються у момент настання подій відповідно до обраних стратегій та тактик. Кожен з них отримує певну подію незалежно один від одного, тобто якщо один з агентів побачив вогонь, а інший ні, та маю стратегію, по якій він не реагує на чужі крики, то для нього подія виникнення вогню не настане.

Подія виявлення вогню потребує реакцію виходу з приміщення, тобто пошуку координат точки виходу за певними критеріями та руху до неї. З огляду на мету системи, ця стратегія є найважливішою для агентів так як неправильний вибір виходу найчастіше призводить до смертельних випадків в реальних ситуаціях.

Усі тактики пов'язані з ним можуть бути задіяні під час руху до виходу та мають на меті допомагати агентові обходити перепони та взаємодіяти з іншими агентами, так як повідомляти оточуючих про виникнення вогню, змінювати їх місце та інше. Дія почати крик також входить к перелік стратегій до даної події. Таким чином можна повідомляти агентів, що в стратегії до подій μπορεί чути панічні крики та реагують на них.

Подія «побачив натовп перед собою» виникає в момент, коли за кілька метрів в напрямку руху або погляду агента з'являється більше ніж 20 інших агентів, що стоять близько один до одного та не змінюють свою позицію так, як вони б це робили під час руху прямо. В цей час агент реагує відповідно до своєї стратегії події руху до виходу, де в деяких її варіантах обирається нова точка виходу якщо між агентом та виходом знаходиться натовп.

Також при цій події та стратегії, що не враховує натовп перед собою може будти задіяна тактика щото реакції на штовхання зі сторони інших агентів. Так, агент при проходженні через натовп може розштовхувати оточуючих у відповідності до цієї тактики, а ті в свою чергу можуть реагувати на нього так само.

Такою тактикою є «Реакція на штовхання», до якої за табл. 1 наразі є два можливих варіантів: відповідати на штовхання або ігнорувати їх та продовжувати рух за обраними стратегіями.

Доцільність та відповідність поведінки агентів реальній буде перевірятися під час реалізації системи на етапі навчання, тому допускається додавати то списку стратегій ті, що на пеший погляд не відтворюють реальну поведінку людей або є неефективними. Таким чином список можливих поведінок збільшиться, що в свою чергу покращить якість на етапі навчання так як обрані стратегії в результаті матимуть обґрунтовану перевагу над іншими.

Тестування та начання буде проходити у середовищі, що має перепони та завади, виходи та місце займання полум'я. Для того щоб точно виявити усі переваги і недоліки різних наборів стратегій та тактик агентів, необхідно дати кожному з них рівні умови, в яких вони можуть зайдіяти цей набір. Тобто стверджувати, що конкретний набір агента не дозволяє йому пристосуватися до оточуючих його подій не є раціональним якщо цей агент з'явиться лише біля місця займання вогню і далеко від виходу, а інший з'явиться лише біля виходу, далеко від вогню та втигне вийти з приміщення до утворення натовпу біля виходів.

Для вирішення цієї проблеми тестування проводиться в декілька ітерацій, коли при кожній з них агенти отримують нову початкову позицію, що було згенеровано навмання а також вогонь з'являтиметься у різних місцях, при цьому повторюючи ці місця для того щоб усі агенти були в приблизно рівних умовах.

### 3.2.2 Розробка критеріїв підбору поведінок за допомогою генетичного алгоритму

Для навчання та підбору оптимальних стратегій відповідно до цілей моделювання, як було вказано вище, застосовано генетичний алгоритм. Завдяки його використанню буде обрано агентів, що виявилися найпристосованішими під час навчальних ітерацій. Кожний набір стратегій агентів не змінюється під час одної фази навчання.

По закінченню такої, найпристосованіші агенти лишаються, а найменш пристосовані видаляються з популяції, де на їхнє місце статують такі набори стратегій, що вбирають в себе частини наборів стратегій найпристосованіших агентів. Обмін та вибірка таких частин відбувається за допомогою мітозу наборів найпристосованіших. Операція мутації в даному випадку не проводиться зовсім по причині того що зміна значення в наборі встановить іншу поведінку, що може відрізнятись від початкої повністю.

Перед тим як визначати найпристосованіших агентів, необхідно визначити самі критерії пристосованості та виділити умови за яких агент може вважатися пристосованим до даної системи. Такі критерії відрізнятимуться в залежності від мети роботи системи.

Далі буде наведено критерії пристосованості для двох систем: тієї, що потребує агентів, котрі лишають приміщення з найменшими жертвами за найменший проміжок часу з пріоритетом найменшої кількості жертв, а також ту, що вимагає певну відсоткову кількість жертв та певний часовий проміжок евакуації.

Так, перша система має на меті виведення агентів, що будуть використовуватися для дослідження найбільш оптимальних стратегій евакуації з приміщення в реальних ситуаціях.

У той час друга імітує реальні дані певних катастроф для навчання популяції агентів, що можуть використовуватися як модель реального, а не ідеального натовпу. Для такої системи такі критерії пристосованості мають бути складнішими для забезпечення не ідеальних, але досі реалістичних агентів.



Отже, для варіанту системи, що вимагає максимальну виживаємість агентів було розроблено наступні критерії пристосованості:

- усього кожне покоління виконується 10 перезапусків середовища. Це дозволить агентам показати свою реальну пристосованість так як вірогідність того, що один агент буде в сумарно в гірших умовах ніж інший, падатиме з кожний перезапуском;

- після 10ти секунд, якщо агент досі знаходиться в приміщенні, він отримує штраф -0.5 до балу пристосованості;

- агент отримує додаткові бали пристосованості якщо знаходиться на відстані меншій ніж 10м від виходу. Максимальна кількість балів – +15;

- якщо агент проходить крізь вихід, для нього дана ітерація завершується, він отримує +30 балів. Враховуючи попередню умову, фактично агент отримає 45 балів за вихід з приміщення;

- при контакті з вогнем агент кожні 200 мс втрачатиме 0.5 бала пристосованості, а при встановленні значення здоров'я на 0 втрачатиме ще 10 балів, що дає в середньому штраф до 25 балів пристосованості. Значення здоров'я досягає 0 при контакті з вогнем сумарно більше ніж 4с;

- при штовханні немає жодних штрафів, проте якщо агента штовхнули, він втрачає 35 пунктів здоров'я. При штовханні інших втрачається 15 пунктів здоров'я. Таким чином якщо два агенти починають штовхати один одного, то після двох поштовхів для них дана ітерація завершується і вони отримують штраф за те, що їхнє здоров'я досягнуло 0;

- нові набори поведінок в подальших перезапусках після 10го генеруються в залежності від загальних показників виживших агентів.

Якщо кількість агентів, що не змогли вийти з кімнати більше ніж 3 рази за один запуск становить більше ніж 10% від усієї популяції, генеруються нові набори поведінки, та замінюються у найменш пристосованих агентів. Кількість заново згенерованих геномів виражається у формулі 3.1, де Р – отриманий відсоток:

$$Q_{10} = Q_{whole} * (P - 10) / 100. \quad (3.1)$$

Усі ці умови системи спонукають генетичний аглорим обирати агентів, що проходять до виходу найбільшу кількість разів та прибирати з популяції тих, хто найбільшу кількість разів потрапляє у полум'я.

У той самий час, визначення критеріїв пристосованості системи, що потребує певної кількості втрат потребує більшої кількості параметрів. Одною зі складностей такого підбору є те, що обрані критерії мають задовольняти більшість можливих вимог до результатів роботи системи. В цьому випадку агенти мають бути навчені поводити себе недостатньо ефективно і досі реалістично.

Для такої системи пропонується наступний ряд правил:

- усього кожне покоління виконується 15 перезапусків середовища;
- після 15ти секунд, якщо агент досі знаходиться в приміщенні, він отримує штраф +3 до балу пристосованості;
- агент отримує додаткові бали пристосованості якщо знаходиться на відстані меншій ніж 10м від виходу. Максимальна кількість балів – +20;
- якщо агент проходить крізь вихід, для нього дана ітерація завершується, він отримує +15 балів. Враховуючи попередню умову, фактично агент отримає 35 балів за вихід з приміщення;
- при контакті з вогнем агент кожні 200 мс втрачатиме 0.2 бала пристосованості, а при встановленні значення здоров'я на 0 втрачатиме ще 10 балів, що дає в середньому штраф до 15 балів пристосованості. Значення здоров'я досягає відразу 0 при контакті з джерелом вогню сумарно більше ніж 6с;
- при штовханні агенти отримують від штрафу до винагороди від -5 до +5 балів в залежності від кількості виживших агентів в минулих двох поколіннях.

Так, після того як агенти починають покидати приміщення у більшій кількості, ніж потрібно, система спонукає їх до штовхани.

Очікується, що агенти завдяки цьому зможуть збалансувати число виживших.

Навчання закінчується якщо на протязі чотирьох перезапусків середня кількість людей, що вийшли дорівнює бажаній з похибкою не більше ніж 5%.

Детальніше про визначення балу пристосованості, мітоз та мутацію описано в наступному підрозділі про навчання агента за допомогою штучної нейронної мережі.

### 3.3 Розробка типів поведінки на основі штучної нейронної мережі

Найефективнішими стратегіями поведінки можуть виявитися ті, що були не запрограмовані раніше, як то набір векторів, за яким агент має рухатися до точки, а ті стратегії, в яких рішення приймаються агентом базуючись за штучній нейронній мережі, що дозволить йому обирати вектор руху самостійно в залежності від навколишніх умов. Таким чином, вже навчений агент поводитиме себе гнучкіше та реалістичніше, що і є метою даної системи. Такі стратегії поведінки будуть додані в систему моделювання нарівні з запрограмованими та обиратимуться генетичним алгоритмом та основі своєї ефективності.

#### 3.3.1 Штучна нейронна мережа, що дозволяє об'єкту дістатися до кінцевої точки, оминаючи вогонь та перепони

Одною з найбільш необхідних стратегій є огинання вогню та інших перешкод, тому в першу чергу саме цю стратегію було побудовано на нейронній мережі.

Було виявлено, що програмування руху агенту під час огинання перепон в даній системі було досить складним та робило дослідженні поведінки

об'єктів в екстремальних умовах важчим так як навчена заново модель може краще висвітлити поведінку, близьку до людської.

Отже для огинання перепон та руху до кінцевої точки агент нейронна мережа повинна мати динамічно обновлювану інформацію про навколишнє середовище, а саме знаходження певних об'єктів навколо нього.

Було введено певну кількість датчиків, що повідомляли цю інформацію: 7 променів від агента, що вимірюють відстань до вогню або стіни, якщо та потрапила у промінь, а також напрямок та відстань до виходу. Довжиною променю було обрано 10м при розмірах агенту 1м, якщо нічого не перетинало промінь, то на вхід подається значення 10, що є максимальним значенням .

Для того щоб зменшити кількість виходів та збільшити точність та зменшити швидкість навчання, вихідними значеннями обрано обертаючи сили агента. Так, сили можуть компенсувати одна одну, даючи змогу агентіві рухатися прямо а також повертати вліво або вправо.

В якості функції активації застосовано біполярну сигмоїду, так як виходи можуть набувати від'ємних значень:

$$f(x) = -1 + \frac{2}{(1+e^{-x})}. \quad (3.2)$$

Входи нейронної мережі (рисунки 3.1):

F (front)– відстань до найближчих об'єктів, що потрапили у вектор  $0^\circ$  відносно напрямку руху агента.

Fr (front left)– відстань до найближчих об'єктів, що потрапили у вектор  $315^\circ$  відносно напрямку руху агента.

Fl (front right)– відстань до найближчих об'єктів, що потрапили у вектор  $45^\circ$  відносно напрямку руху агента.

L (left)– відстань до найближчих об'єктів, що потрапили у вектор  $270^\circ$  відносно напрямку руху агента.

R (right)– відстань до найближчих об'єктів, що потрапили у вектор  $90^\circ$  відносно напрямку руху агента.

BR (back right)– відстань до найближчих об'єктів, що потрапили у вектор  $135^\circ$  відносно напрямку руху агента.

BL (back left)– відстань до найближчих об'єктів, що потрапили у вектор  $225^\circ$  відносно напрямку руху агента.

$\alpha$  – кут між вектором front агента та напрямком до кінцевої точки.

$d$  – відстань від агента до кінцевої точки.

Виходи нейронної мережі (рисунок 3.1):

Right rotate force – сила з якою агент обертається вправо.

Left rotate force – сила з якою агент обертається вліво.

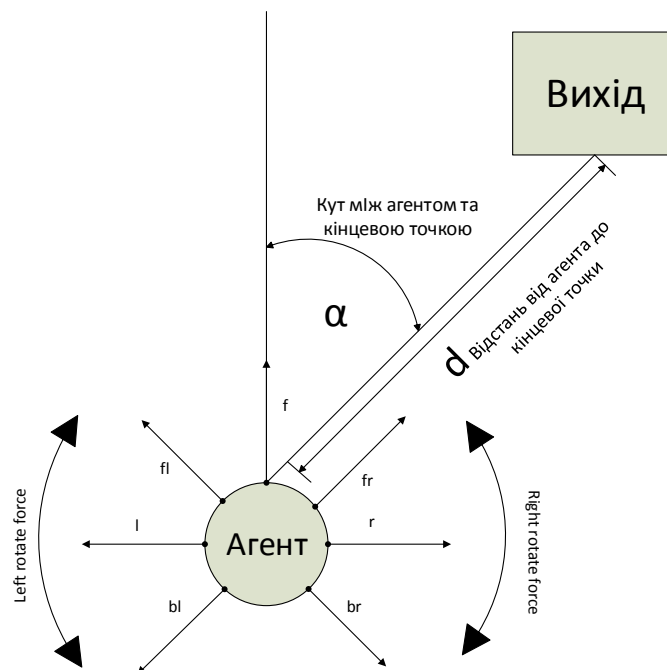


Рисунок 3.1 – Входи та виходи агента

В нейронній мережі окрім вхідного та вихідного шарів побудовано один прихований шар, що має 10 прихованих нейронів (кресленик ІА.61.140БАК.001 Д1).

Далі буде описано побудову середовища для навчання:

Для навчання агентів було побудовано тривимірну модель кімнати зі стінами та виходами як вказано на рисунок 3.3.

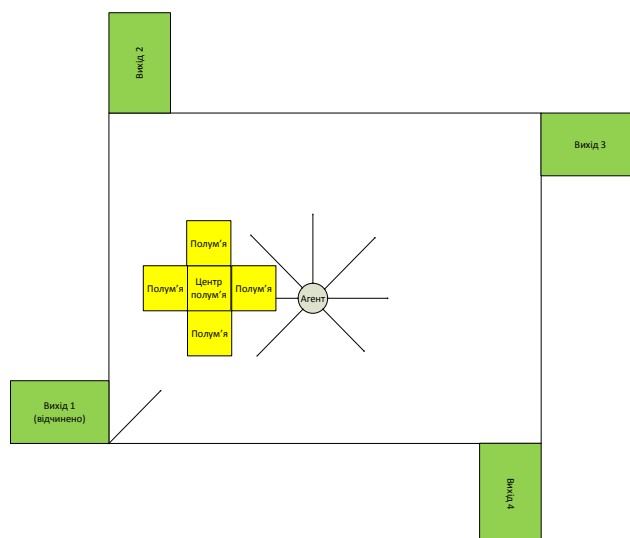


Рисунок 3.2 – Варіант розташування об’єктів у тренувальній кімнаті

Кожен раз на початку полум’я з’являється в певній точці та розгорається радіально навколо до того часу поки не займе всю кімнату, якщо за цей час агент не дістався виходу, він не має можливості не потрапити у вогонь. Агент має число, що виражає його здоров’я та має початкове значення 100. При знаходженні агента у полум’ї, число здоров’я знижується і при досягненні цього значення до 0 агент завершує ітерацію навчання.

Навчання кожного з геномів поведінки агента проходить таким чином, що всі агенти певного покоління навчання мають одні й ті самі умови кімнати, а саме позицію, обернення по осі Z, місце початку полум’я та вхід, що відкрито. Кожну з навчальних ітерацій агент з’являється в новому місці та записує його для наступних геномів.(таблиця 3.2) Такий підхід застосовано для кращого виявлення найпристосованішого геному.

Таблиця 3.2 — Приклад умов що має агент для різних ітерацій

№	Позиція агента	Рот. агента по осі Z	Центр полум’я	Вхід відкрито
1	x1, y1	z1	v1, w1	перший
2	x2, y2	z2	v2, w2	другий

3	$x_3, y_3$	$z_3$	$v_3, w_3$	четвертий
4	$x_4, y_4$	$z_4$	$v_4, w_4$	третій
5	$x_5, y_5$	$z_5$	$v_5, w_5$	другий

### 3.3.2 Навчання мережі генетичним алгоритмом

Внутрішньо нейронні мережі представлені у вигляді послідовності числових матриць ваг кожного шару окрім вхідного. Загалом ці матриці мають вигляд схожий до того, що описано, додавання та видалення нейронів також відбувається відповідно [11].

Отже перелік всіх ваг між вхідними, прихованими та вихідними нейронами є геном поведінки агента. На рисунку 3.4 вказано як саме ваги починаючи з першого вхідного нейрону та закінчуючи останнім вихідним записуються по порядку.

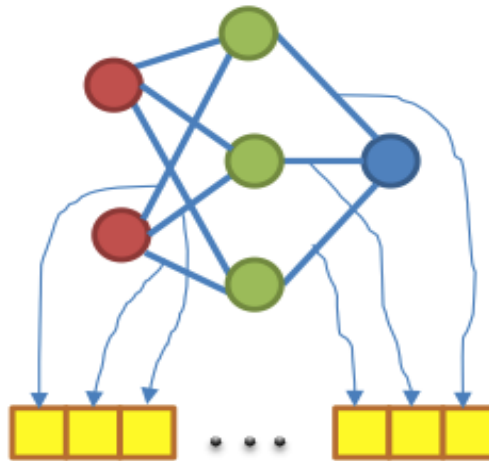


Рисунок 3.3 – Вигляд запису геному агента [12]

Найважливішим показником того, що саме цей геном має бути обраним для майбутніх поколінь є пристосованість (fitness) геному.

Так як головною метою агента є знаходження виходу, а при попаданні у полум'я він зазнає невдачі, то до значення фітнесу додаватиметься 30 при

знаходженні виходу, натомість при попаданні в полум'я, кожні 200 мс агент разом зі значенням здоров'я втрачатиме 0.5 бала пристосованості, а при встановленні значення здоров'я на 0 втрачатиме ще 5 балів, що дає в середньому штраф до 15 балів пристосованості. Також штраф накладається на агента якщо він надовго затримується у кімнаті. Починаючи з 5 секунд знаходження в кімнаті агент отримує штраф в 1 бал на секунду. На початку ітерації агент має початковий бал пристосованості 15. При завершенні ітерації отриманий бал додається до сумарного балу пристосованості поточного геному.

При першому запуску у навчанні приймають участь 30 геномів, зі значеннями ваг генерованих навмання. Кожен з них проходить 5 ітерацій навчання, після чого сумарний геном записується до бази даних. Далі всі наступні геноми проходять ті ж ітерації доки вони не закінчаться. По завершенню навчання поточного покоління обираються 5 найбільш пристосованих геномів, що породять наступне покоління. Наступні геноми лишаються в популяції та утворюють між собою свої змінені копії шляхом мітозу як показано на рисунку 3.4. Обидва геноми розділяються на певній мітці та обмінюються частинами генетичного коду утворюючи при цьому два інші геноми, що наслідуватимуть частини батьківських. На лістингу 1 висвітлено функцію мітозу.

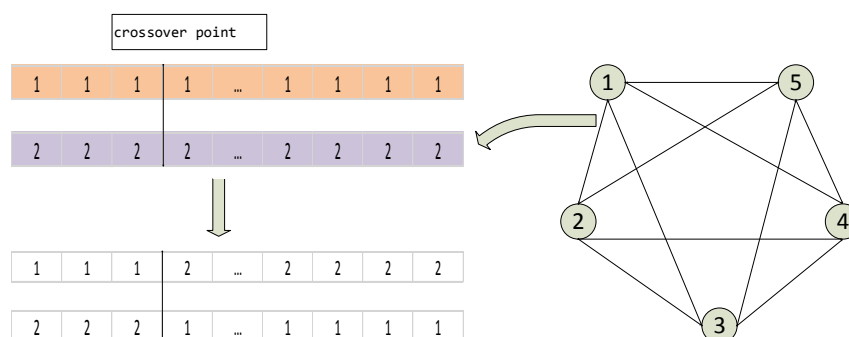


Рисунок 3.4 – Мітоз між найпристосованішими геномами

```
void CrossBreed (Genome g1, Genome g2, ref Genome baby1, ref Genome baby2){
```



```

int totalWeights = g1.weights.Count;
int crossover = Random.Range(0,totalWeights-1);
baby1 = new Genome (); baby1.ID = genomeID;
baby1.weights = new List<float> ();
for(int i=0; i<totalWeights; i++)
{baby1.weights.Add (0.0f);}genomeID++;
baby2 = new Genome ();baby2.ID = genomeID;
baby2.weights = new List<float> ();
for (int i=0; i<totalWeights; i++)
{baby2.weights.Add (0.0f);}genomeID++;
for(int i=0; i< crossover; i++){
baby1.weights[i] = g1.weights[i];
baby2.weights[i] = g2.weights[i];}
for(int i = crossover; i< totalWeights; i++){
baby1.weights[i] = g2.weights[i];
baby2.weights[i] = g1.weights[i];}

```

### Лістинг 1 – Фрагмент коду, що здійснює мітоз

Кожен геном після цього проходить операцію мутації, де кожен його ген, тобто вага зв'язку в нейронній мережі, можуть бути змінені навмання з вірогідністю `MUTATION_RATE` на випадкове значення в діапазоні `MAX_PERBETUATION`

```

void Mutate(Genome genome){
for (int i=0; i<genome.weights.Count; i++) {
if(RandomClamped() < MUTATION_RATE){
genome.weights[i] += (RandomClamped() ° MAX_PERBETUATION);}}}

```

### Лістинг 2 – Фрагмент коду, що здійснює мутацію

Так утворюються перші 25 геномів наступного покоління. Решта 5 генеруються навмання для підвищення різноманітності у популяції.

Навчання завершиться коли один і той же геном пройде поріг достатнього балу пристосованості, тобто знайде вихід принаймні 13 разів (4.3 рази за одну епоху), на протязі трьох поколінь.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОЇ СИСТЕМИ

### 4.1 Оцінка ефективності запропонованої системи та її програмної реалізації

В даному розділі описано реалізацію обох систем: тої, що моделює поведінку натовпу вцілому і реалізацію навчання поведінки на штучній нейронній мережі.

Детально пораналізовано роботу генетичного алгоритму як підходу навчання при різних налаштуваннях для обох розглянутих систем.

#### 4.1.1 Опис реалізації системи вцілому

Для реалізації завдання створення агентної системи моделювання поведінки натовпу була використана мова програмування C# та тривимірний рушій Unity3d.

Основним критерієм вибору цього фреймворка була можливість швидкого створення тривимірних об'єктів на додавання їх на основну сцену, набір інструментів, що спрощують програмування руху тривимірних об'єктів, підтримка мов об'єктно-орієнтованого програмування, таких як C# та JavaScript. Та, власне, логіка виконання скриптів, коли до тривимірного об'єкту на сцені можливо динамічно додавати класи, що будуть виконуватися лише даним об'єктом. Діаграма класів представлена в кресленику ІА.61.140БАК.001 Д2.

В якості технології збереження даних було використано базу даних SQLite по причині того, що вона добре підходить в якості локальної бази даних, швидко підключається та має поширення на багатьох платформах. Основною метою бази даних було збереження відомостей про набори геномів агентів та налаштування зв'язків між стратегіями.

Програма складається з класів управління, моделей агента і геному та частини для генетичного алгоритму. Також було створено нейронну мережу, що задіює у собі класи нейрону, синапсу, мережі. У програмі використано патерн програмування Стратегія так як він найбільш вдало допомагав додавати певні стратегії поєдинки тривимірним моделям агентів.

Також програма побудована таким чином, що можливо додавати нові стратегії поведінки або події і такі дії не будуть заважати вже створеним поведінкам завдяки даному патерну.

Стратегії поведінки представляють собою окремі скрипти, що додаються до об'єктів агентів за допомогою класу DB\_uploader, що завантажує необхідні зв'язки між певними агентами та стратегіями поведінки з бази даних. Стратегії обходу перешкод представлені в кресленику ІА.61.140БАК.001 Д1.

При першому запуску програми відбувається наповнення геномів за допомогою класу GenomeSpawner, де до бази даних записуються зв'язки між агентами та стратегіями.

На рисунку 4.1 наведено приклад такого наповнення. Відповідно до ідентифікаційного номеру агента, його геном записується у вигляді номерів ідентифікаторів стратегій на певні події.

Ці поля мають атрибут FOREIGN KEY, що пов'язує ідентифікатори в таблиці агентів з ідентифікаторами назв стратегій. Саме це забезпечує не тільки безпечний зв'язок, а й гарантує, що при зміні, видаленні або редагуванні ідентифікатора в таблиці певних стратегій, ідентифікатор в таблиці агентів також зміниться на відповідний, що унеможлиблює й приводить до мінімуму ситуації, коли виникає посилання на хибні або неможливі стратегії.

На рисунку 4.1 видно розподілення за трьома стратегіями та тактиками: помітити вогонь, помітити крики, реакція на штовхання.

Так, по події OnTriggerEnter агент отримує дані про інший об'єкт, що знаходиться в його «зоні видимості» і якщо цей об'єкт є вогнем, то спрацьовує подія «помітив вогонь» після якої вмикається відповідна стратегія.

id	NoticeFireStrat	NoticeScreamingStrat	PushReactionStrat
1	2	1	1
2	1	2	2
3	1	2	2
4	2	1	2
5	1	1	1
6	1	1	2
7	1	1	2
8	2	2	1
9	2	1	2
10	2	1	2
11	1	2	1
12	1	1	2
13	1	1	2
14	1	1	2
15	1	1	1
16	1	2	1
17	2	2	1
18	1	1	1
19	1	1	1

Рисунок 4.1 – Приклад наповнення геномів агентів

Сам агент має базовий скрипт Scene\_Human, що має властивості класу Human, тобто максимальний запас здоров'я, ідентифікатор геному агента, властивості пов'язані з рухом, такі як швидкість, номери ідентифікаторів стратегій та базові сенсори, що забезпечують розпізнавання певних подій. Тривимірні агенти мають на собі фізичний компонент Unity Collider. Фактично він є тривірним полем у просторі, що може перевищувати об'єм самого агента та слугує простим сенсором навколишнього середовища, завдяки якому можна отримати дані про інші колайдери, що перетнули його.

#### 4.1.2 Дослідження роботи системи моделювання натовпів

Було реалізовано обидві ситуації, описані в попередньому розділі: систему, що потребує найкращих агентів та вихід усіх агентів за якомога короткий

проміжок часу, та ту, що має моделювати ситуацію з заданими агентськими жертвами.

В ході обробки результатів реалізації було помічено, що після введення тактики обходу вогню за допомогою нейронної мережі, агенти, що обирали її набагато краще справлялися з цим завданням і з плином часу цей ген став домінуючою стратегією.

Цікаво, що перша система доволі швидко та ефективно справилася із завданням виведення популяції, що залишає приміщення якомога швидше та за найкоротший проміжок часу.

Також відмічено, що для досягнення поставленої мети агенти не прийняли абсолютно однакові геноми, частина генів виявилася спільною для більшості, проте деякі з них відрізнялися і щодо кількох стратегій зовсім не було виділено домінуючих.

Так, кінцевими генами поведінки реакцій на виявлення вогню були всі наведені, окрім тої, що спонукає агентів рухатися до головного виходу, незважаючи ні на що. Також було виділено різні типи поведінок по відношенню до штовхання.

Лише 35% з фінальних агентів штовхали у відповідь тих, хто зробив це першим.

Знайдено закономірності між геномами агентів. Так, агенти розділилися на “агресивних”, “спокійних” та “неявних”, тобто таких агентів, що пристосувалися без явних закономірностей. Так, агресивні агенти реагували на штовхання і не змінювали напрямку руху, якщо побачили натовп, таких агентів у популяції було близько 33%.

В свою чергу спокійні агенти не відповідали на штовхання і обирали інший шлях перед натовпом біля дверей, але поверталися, якщо позаду був натовп, реагували на крик. Таких спокійних агентів було близько 40% у фінальній популяції.

Закономірності з руху до вогню не було виявлено, скоріш за все по причині того, що якщо б наприклад усі агресивні агенти рухалися до одного виходу, то вони влаштували б штовханину і навпаки, деякі агенти виходили кріз вогонь з приміщення якомога швидше і отримували свої бали.

Приблизне відсоткове співвідношення обраних генів можна побачити в таблиці 4.2.

Таблиця 4.2 – Результати навчання при різних конфігураціях

Тип руху	Виявлення вогню	Почув панічні крики	Побачив натовп перед собою	Реакція на штовхання
Рух до цілі незважаючи ні на що <b>1%</b>	Рухатися прямо до найближчого головного виходу, незважаючи ні на що <b>5%</b>	Почати стратегію в виявленні вогню <b>66%</b>	Обрати інший шлях виходу <b>25%</b>	Ігнорувати <b>65%</b>
Перед перепорою змінювати вектор руху на 30° <b>4%</b>	Рухатися до найближчого виходу, що помітив, почати крик <b>39%</b>	Ігнорувати крики <b>33%</b>	Проходити ігноруючи натовп <b>40%</b>	Штовхнути у відповідь <b>35%</b>
Рух на основі стратегії нейронної мережі <b>95%</b>	Рухатися до найближчого головного виходу, якщо той не зайнявся полум'ям <b>56%</b>		Обрати інший шлях виходу, ігнорувати, якщо позаду вогонь <b>35%</b>	

Реалізація системи із заданими втратами серед агентів за умовами, вказаними в минулому розділі, не була вдалою. Агенти не встигали навчитися

достатньо і замінювалися новими. При досягненні фінального покоління система не виглядала такою, що в повній мірі хоча б приблизно імітує поведінку натовпу. Було вирішено замінити правила вибору агентів та ті, що були в попередній системі та просто зупиняти її навчання в момент коли кількість агентів, що вийшли на протязі трьох поколінь становить певну частину. Незважаючи на нормальні результати, така система поки не рекомендується до використання так як деякі з агентів все одно не показували бажаної поведінки.

#### 4.1.3 Реалізація алгоритму навчання модулю на нейронній мережі

Реалізована програмна система була також використана для дослідження та тренування моделі агента обходити перешкоди та діставатися до кінцевої точки. Було розглянуто роботу нейронної системи з різною структурою: один запуск відбувався з характеристиками, наведеними вище, в іншій конфігурації системи вихід був лише один та залишався на місці, агент мав лише 4 промені-датчика, а за знаходження більше 5ти секунд на полі надавалася винагорода. Структура представлена у креслинику ІА.61.140БАК.001 Д1.

Додатково до програмної реалізації, описаної раніше, реалізації променів було використано метод RayCast, що випускає промені у напрямку вектора та на певну довжину. При попаданні в інший колакйдер, він видає інформацію по його власника. В нашому випадку це може бути вогонь або інший агент.

#### 4.2 Дослідження роботи алгоритму

Було виявлено, що кількість входів та якість тренувального середовища значно впливають на швидкість та ефективність навчання.

Було розроблено та побудовано найбільш доречне тренувальне середовище, що дозволяло досягти найбільш точного та універсального результату з навчання агента оминати перепони у просторі.

Для системи наведеної вище, геном, що здатен знаходити вхід 13 разів за 3 покоління підряд з'являвся в середньому на 21ше покоління, а отже такий геном було виведено. Можемо зробити висновок, що вже на 18му агенти починали в своїй більшості рухатися завжди у сторону виходу. При цьому можна побачити, що вони рухались в напрямку вихода починаючи з 6-7го покоління, при цьому маючи середній рівень пристосованості 130, а максимальний 170.

Після того як штраф за знаходження на полі більше 5 секунд замінили на приз відстані від виходу, тобто чим ближче до виходу знаходився агент, тим більше балів він отримував, фінальний геном формувався вже на 15му поколінні в середньому. Саме через зміну розбаловки агенти навчилися рухатись до виходу швидше.

В свою чергу спрощена система набагато швидше досягала фінальних результатів, близько 11го покоління вже було сформовано фінальний ген, що міг проходити задану кількість виходів за 3 покоління.

Модель, що мала лише 4 входи, промені лише за кутами  $0^\circ$ ,  $315^\circ$ ,  $45^\circ$  та  $180^\circ$  мали на початку менше середнє значення пристосовності, натомість формували фінальний геном вже до 7-8го покоління.

Проте дані моделі були неспроможні та нездатні поводити себе вірно в інших змодельованих ситуаціях, окрім конкретної навчальної, в якій вони навчалися.

Якщо вихід пересувався або їхня ротація в просторі змінювалася від ітерації до ітерації, було видно, що вони мали помилкову стратегію поведінки та не могли досягти результату.

Рисунок 4.2 ілюструє різну поведінку, що показали геноми агентів в різних середовищах.



Зміна місця виходів допомагала агентам побудувати чітку залежність між кутом та відстанню до виходу та балом пристосованості.

Ті що знаходилися в незмінній кімнаті досить швидко почали виходити в кімнати, проте їхні нейронні мережі базувалися лише на входах, що приходили зліва та завжди обходили всю кімнату за годинниковою стрілкою тримаючись при цьому стіни. Блок-схема роботи алгоритму представлена в креслинику ІА.61.140БАК.001 ДЗ.

З рисунку нижче видно, що перша мережа більш гнучка та ефективна через те що агенти приділяли увагу не тільки оминанню перешкод, а й напрямку виходу.

Це відбулося завдяки тому що його положення також змінювалося.

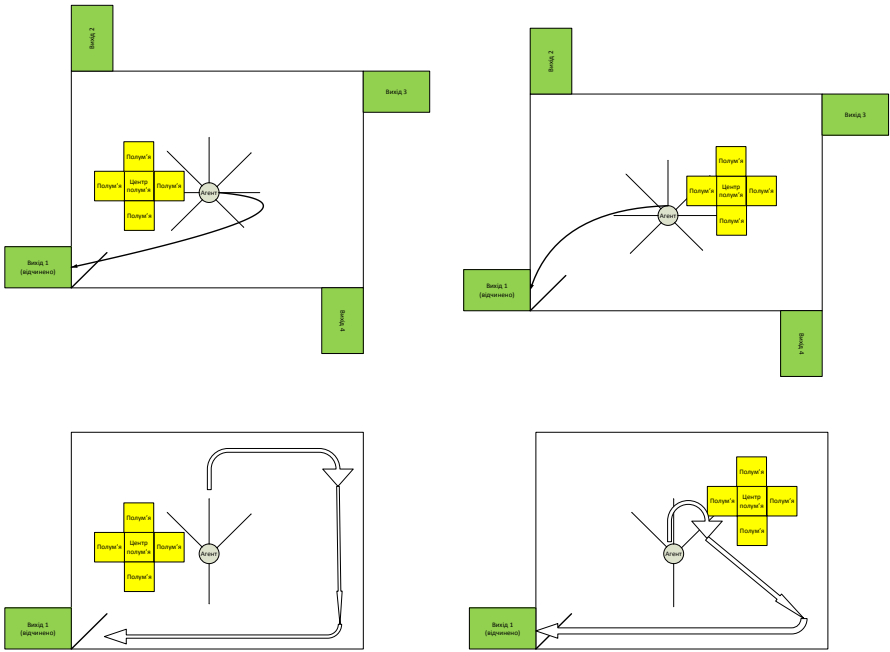


Рисунок 4.2 – Маршрут руху найбільш пристосованого агента в різних середовищах

На таблицях 4.3 та 4.4, вказано середній (а) та найвищий (б) бали пристосованості геномів в різних поколіннях. Значення вказані у відсотках до максимально можливого в даній конфігурації.

Як видно з результатів, для досягнення певної цілі важливо мати достатньо повні навчальні фактори середовища.

Зрештою було побудовано таку конфігурацію мережі та середовища, при якій агент показував бажану поведінку навіть при зміні середовища.

Таблиця 4.3 – Результати навчання при різних конфігураціях

Конфігурація системи \ № покоління	3	6	9	13	16	19	21
4 промені, 1 вихід, без штрафу за відстань	a- 30 b- 63	a- 45 b- 69	a- 53 b- 70	a- 44 b- 68			
4 промені, 1 вихід, штраф за відстань	a- 34 b- 66	a- 40 b- 75	a - 55 b- 89	a - 53 b- 95	a- 50 b- 83	a- 55 b- 75	a- 57 b- 81
9 променів, 1 вихід, штраф за відстань	a- 40 b- 72	a- 45 b- 80	a- 57 b- 93	a- 56 b- 95	a- 59 b- 95	a- 60 b- 94	a- 65 b- 95
9 променів, 4 виходи, без штрафу за відстань	a- 25 b- 65	a- 43 b- 70	a- 45 b- 73	a- 51 b- 75	a- 50 b- 83	a- 50 b-92	a- 50 b- 95
9 променів, 4 виходи, штраф за відстань	a- 23 b- 80	a- 32 b- 80	a- 55 b- 74	a- 56 b- 78	a- 62 b- 94	a- 60 b- 96	a- 64 b- 93

Таблиця 4.4 – Досягнення фінального геному при різних конфігураціях

Конфігурація системи \ № покоління	Покоління форм. Фін. Геному	Підходить для інших середовищ
------------------------------------	-----------------------------	-------------------------------

4 пр., 1 вих., без штрафу	-	ні
4 пр., 1 вих., штраф	13	ні
9 пр., 1 вих., штраф	9	ні
9 пр., 4 вих., штраф	18	так
9 пр., 4 вих., штраф	15	так

#### 4.3 Пропозиції щодо програмної реалізації системи моделювання поведінки натовпу під час пожежі

Проведення експериментальних досліджень показало, що збільшення переліку можливих подій стратегій та повторів добре впливає на роботу системи вцілому та розширює можливі комбінації поведінок, що в свою чергу робить популяцію більш адаптованою. Діаграма варіантів використання системи представлено в креслинику ІА.61.140БАК.001 Д4. Також розроблена та навчена штучна нейронна мережа показала кращі результати у порівнянні з застріптованими стратегіями поведінки.

Недоліками цієї роботи є поки що недостатня реалістичність поведінки натовпу, так як складно змодельовати повністю ідентичну людському характеру та рисам програму, особливо коли на меті системи є утворення популяції агентів, де деякі з них мають загинути та бути недосатньо адаптованими до виходу. Для такої реалізації необхідно додати більше можливостей агентам, доповнити критеріями їх поведінку та фізичні можливості. Змодельовати та розробити таку методику навчання, за якої контроль агентів відбуватиметься точніше і оброюлятися більш детально. Для цього необхідно ускладнити середовище навчання додатковими та новими перешкодами, додати нових агентів, що допомагали б іншим виходити з приміщення і цим ускладнити забезпечення виходу всіх агентів.

З огляду на все це, пропонується розширити список можливих подій та створення відповідних стратегій та тактик з поглибленням у такі, що виражають поглиблення взаємодії між агентами в психоемоційному плані, виділення рівня агресії у агентів, авторитету та довіри до інших, зробити систему максимально схожою до емоцій та дій реального натовпу. Так, якщо агент довіряє іншому агентові з більшим рівнем авторитету, то починає повторювати дії за ним. Таким чином в натовпі виділятимуться лідери, що керуватимуть натовпом.

Рекомендується додати більше стратегії, створених за допомогою нейронних мереж, так як такі будуть особливо високоефективними в стратегіях спілкування. Також потрібно по можливості побудувати хоча б дві нейромережеві стратегії, що мали різне навчання до кожної з представлених подій, по можливості з різною структурою, з метою дослідження роботи цих структур для навчання агентів.

Додавання нових агентів, що мають інші ролі, приблизить моделювання до реалістичного. Можливе додавання агентів, що імітують працівників правопорядку, медиків, пожежників. Такі агенти матимуть нові функції, їх система накопичування балів має бути переглянута. Також вони мають вищий рівень авторитету та великий вплив на натовп.

Іншим підходом до майбутніх доопрацювань може бути розширення роботи нейронних мереж та навчання генетичним алгоритмом до побудови кроків під час стратегій поведінки. Тобто при прийнятті рішення об'єктом він сам обиратиме собі стратегію поведінки з коротких рухів, тобто формуватиме особистий маршрут поведінки. Це дозволить показати моделювання більш реально, та не просто виявити можливі найбільш ефективні моделі поведінки об'єктів в екстрених ситуаціях, а й побудувати відповідні.

## ВИСНОВКИ

В дипломному проекті було створенно методику та систему, що навчає та моделює агентну поведінку під час виходу агентів з палаючого приміщення в якому є перешкоди та пожежний вихід. Для вирішення завдання було проведено аналіз та порівняння різних видів агентних моделювань, навчання агентів та нейронних мереж, їх поєднань, а також визначено їх переваги і недоліки та розроблено алгоритми тренування та моделювання. Для досліджень було порівняно різні системи агентного моделювання, зокрема моделювання поведінки натовпів, виділено їхні недоліки та запропоновано нову методику, що центрується на поведінці агентів саме під час екстремальних подій. В якості методу навчання обрано генетичний алгоритм, що здатен проводити навчання без учителя та правильних відповідей та є гарним для систем та ситуацій без очевидних відповідей і соціальних явищ.

Основною причиною того, що було обрано саме генетичний алгоритм, є те, що наколишнє середовище може змінюватися.

За результатами такого можна фактично експериментально отримати результуючі локальні та можливо глобальні мінімуми функцій для ефективного виходу кожного типу особистості об'єкту та груп об'єктів в цілому. Дані значення дуже важливі для досліджень групової поведінки так як задача з побудування самої функції є дуже складною.

Проведено детальний аналіз роботи системи, виділено недоліки та у зв'язку з ними додано пропозиції для розвитку розробляємої системи.

Також побудована система ефективна тим, що додавання додаткових станів дій розробником відбувається досить швидко. Таким чином майбутні версії застосунку можуть змінювати форму та зміст не руйнуючи та не змінюючи основну архітектуру, при цьому виконуючи інші функції та набуваючи розширень.

Отримані результати можна використати в інструктажах з техніки безпеки при екстрених ситуаціях, таких як пожежі, також порівняти поведінку моделей на певному етапі навчання з поведінкою реальних натовпів та виявити моделі, що підходять для імітації людської поведінки. Подібну методику та програму можна використовувати при проектуванні торгівельно-розважальних центрів або кінотеатрів, щоб вирішити, де розмістити пожежний вихід.

Так, при зміні навколишнього середовища, та зберігаючи певні критерії його обладнання можливо розширити дане застосування до такого, що дозволить перевірити доцільність форми та плану будівель щодо можливості заявленої кількості людей евакуюватися з нього з мінімальними втратами у найшвидший термін.

					ІА61.140БАК.005 ПЗ	Лист
Змн.	Аркуш	№ докум.	Підпис	Дата		64

## ПЕРЕЛІК ПОСИЛАНЬ ТА ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коллективное переживания. Натовп. [Электронный ресурс] / Douglas Eck – Режим доступа до ресурсу: [http://megalib.com.ua/content/3242\\_Kolektivne\\_perejivannya\\_Natovp.html](http://megalib.com.ua/content/3242_Kolektivne_perejivannya_Natovp.html).
2. Назаретян А. П. Психология стихийного массового поведения. Лекции»: ПЕР СЭ; М.; 2001
3. NetLogo is a multipagent programmable modeling environment. [Электронный ресурс] – Режим доступа до ресурсу: <https://ccl.northwestern.edu/netlogo/>
4. Swarm Development Group [Электронный ресурс] – Режим доступа до ресурсу: [http://www.swarm.org/wiki/Main\\_Page](http://www.swarm.org/wiki/Main_Page)
5. Solutions in Pedestrian Dynamics company. Airport crowd simulations. [Электронный ресурс]. Режим доступа: <http://www.pedestrian-dynamics.com/crowd-management/airport-crowd-simulation.html>
6. Szymanczyk, Oliver and Duckett, Tom and Dickinson, Patrick (2012) Agent-based crowd simulation in airports using games technology. In: Transactions on computational collective intelligence. Lecture Notes in Computer Science. [Электронный ресурс]. Режим доступа: <http://eprints.lincoln.ac.uk/6574/>
7. Michael Nielsen. Neural Networks and Deep Learning [Электронный ресурс] / Michael Nielsen – Режим доступа до ресурсу: <http://neuralnetworksanddeeplearning.com/>.
8. Martin Gardner The fantastic combinations of John Conway's new solitaire game «life» // Scientific American. — № 4 (October 1970).
9. Искусственная жизнь. Генетический алгоритм v2.0. Первые пробы. [Электронный ресурс]. Режим доступа: <https://www.youtube.com/watch?v=PCx228KcOow/>

10. Дивеев А. И., Софронова Е. А. “Основы генетического программирования Учебно-методическое пособие” - М.: Изд-во РУДН, 2006.
11. Дорогий Я. Ю. Реалізація алгоритму структурної оптимізації нейронної мережі / Я. Ю. Дорогий, О. О. Дорога-Іванюк, Д. А. Ференс // Вісник Національного технічного університету України «КПІ». Інформатика, управління та обчислювальна техніка. – 2015. – Вип. 63. – С. 101-106. - Режим доступу: [http://nbuv.gov.ua/UJRN/Vkpi\\_iuot\\_2015\\_116](http://nbuv.gov.ua/UJRN/Vkpi_iuot_2015_116).
12. Yurii Lahodiuk. Эволюция агентов, управляемых нейронной сетью. [Электронный ресурс]. Режим доступу: <https://habrahabr.ru/post/168067/>
13. Минский М., Пейперт С. Перцептроны. М.: Мир, 1971 — 261 с.
14. Xabier Basogain Olabe. Redes Neuronales Artificiales y sus Aplicaciones Formato Impreso: Publicaciones de la Escuela de Ingenieros, 1998 – 79 p.
15. Дивеев А. И., Софронова Е. А. “Основы генетического программирования Учебно-методическое пособие” - М.: Изд-во РУДН, 2006;
16. Hadzibeganovic, Tarik; Stauffer, Dietrich; Schulze, Christian (2009). «Agent-based computer simulations of language choice dynamics». Annals of the New York Academy of Sciences. 1167: 221–229. [Электронный ресурс]. Режим доступу: <http://adsabs.harvard.edu/abs/2009NYASA1167..221H>
17. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. — М.: Финансы и статистика, 1989. — 607 с.
18. Светульников И.С., Светульников С.Г. Методы социально-экономического прогнозирования. Том 1. Теория и методология. — 1. — Москва: Юрайт, 2015. — С. 30. — 351 с.
19. Дж., Гонсалес Р. Принципы распознавания образов, М. 1978.
20. Терехов В. А., Ефимов Д. В., Тюкин И.Ю. Нейросетевые системы управления: Учеб. Пособие для вузов — М.: Высш. шк. 2002. — 183 с.: ил.
21. Хайкин, С. Нейронные сети: полный курс, 2-е изд., испр. : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2006. — 1104 с.